
Universidade de Aveiro
2008
Departamento de Electrónica, Telecomunicações e Informática

Nuno Rui Cardoso Faria Guimarães

Mineração de texto em literatura biomédica

Mineração de Texto em Literatura Biomédica

Universidade de Aveiro
2008
Departamento de Electrónica, Telecomunicações e Informática

Nuno Rui Cardoso Faria Guimarães

Mineração de texto em literatura biomédica

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e de Telecomunicações, realizada sob a orientação científica do Dr. José Luís Oliveira, Professor Associado do Departamento de Engenharia Electrónica e de Telecomunicações da Universidade de Aveiro

Dedico este trabalho aos meus pais, à minha esposa e ao meu filho pelo incansável apoio e pela infinita paciência.

O júri

Presidente

Dr. Joaquim Arnaldo Carvalho Martins
Professor Catedrático da Universidade de Aveiro

Dr. Gabriel de Sousa Torcato David
Professor Associado do Departamento de Engenharia Informática
da Faculdade de Engenharia da Universidade do Porto

Dr. José Luís Guimarães Oliveira (orientador)
Professor Associado da Universidade de Aveiro

Agradecimentos

Agradeço a todos aqueles que sempre me deram força para continuar.
Um obrigado especial a todo o grupo de bioinformática.

Palavras-chave

Mineração de texto, recolha de documentos, selecção de atributos, modelo vectorial de representação de documentos, similaridade, literatura biomédica

Resumo

O presente trabalho apresenta as técnicas e metodologias principais envolvidas nas diversas fases que constituem um sistema de mineração de texto. É descrito o desenvolvimento de três aplicações que permitem a construção de representações de corpus de documentos da área da biomedicina, e a recolha de documentos similares a partir de outra corpora. São descritas as experiências efectuadas com estas aplicações e são discutidos os resultados obtidos.

Keywords Text mining, information retrieval, feature selection, vector space model, similarity, biomedical literature

Abstract This work presents the main techniques and methodologies involved on the different phases of a text mining system. It is described the development of three applications that manage the generation of representations for biomedical corpus and the retrieval of similar documents from other corpora.
The experiences made with these applications are also described and the obtained results are discussed.

Índice

1. Introdução

1.1. Enquadramento e objectivos

Com o aumento rápido do volume de informação científica presente em texto (nomeadamente em formato digital) torna-se difícil aos investigadores conseguirem acompanhar e explorar todo o conhecimento disponível mesmo quando se trata de nichos ou subdomínios biomédicos. O tamanho de colecções de documentos como a PubMed [1], por exemplo, que contém os resumos (*abstracts*) de mais de 12 milhões de documentos, torna impossível qualquer tentativa manual de correlação de informação. Deste modo, é desejável que exista uma forma automática, computorizada, que permita não só a selecção e recolha dos documentos que podem eventualmente ser de interesse para uma área específica, mas também que consiga explorar o conhecimento subjacente, aumentando a rapidez e eficiência das actividades de investigação. As técnicas e metodologias envolvidas na mineração de texto permitem que os investigadores tenham disponíveis ferramentas que executam essas tarefas, existindo consequentemente todo o interesse em explorar as suas potencialidades e estudar a sua aplicação no domínio da biomedicina. Nesta dissertação pretendeu-se reunir as noções principais sobre mineração de texto, apreender do actual estado da arte e aplicar técnicas, metodologias e algoritmos no desenvolvimento de algumas ferramentas.

1.2. Estrutura da dissertação

A estrutura desta dissertação encontra-se basicamente alicerçada em duas zonas principais: uma de apresentação de metodologias, técnicas e estado da arte da mineração de texto, de que consta o capítulo 2 e seus subcapítulos; e uma de descrição de teor eminentemente prático, em que se descrevem as ferramentas desenvolvidas e as experiências efectuadas, que compõe o capítulo 3 e seus subcapítulos.

No capítulo 2 começam-se por apresentar a definição de mineração de texto e as fases principais que fazem parte de um processo genérico de mineração, nomeadamente um pré-processamento (ao qual é dado um ênfase particular), a aplicação de algoritmos nucleares de mineração, a visualização e um pós-processamento. A descrição detalhada destas diferentes fases e das actividades que as constituem acabam por constituir os diversos subcapítulos.

Assim, no ponto 2.1. é abordada a fase de pré-processamento, fundamental para o sucesso de um processo de mineração, pois é neste passo primário que se tenta estruturar

da forma mais adequada a informação obtida dos textos para uma posterior aplicação eficiente dos algoritmos de mineração. São descritas as técnicas de processamento preparatório de documentos (que tentam adicionar uma primeira estruturação simples, básica, aos documentos – ponto 2.1.1.), de processamento de linguagem natural (que permitem uma análise lexical e sintáctica da informação textual – ponto 2.1.2.), de selecção de atributos e de modelos de representação para os documentos (que transformam os documentos em representações que têm o objectivo de ajudar a melhorar o desempenho dos sistemas e facilitar a aplicação de diversos algoritmos – ponto 2.1.3.), de recolha de documentos de interesse e classificação (que resultam em subconjuntos semanticamente relacionados de documentos – ponto 2.1.4.) e de extracção de informação (que extraem entidades e factos que envolvem essas entidades, organizando a informação em unidades estruturadas – ponto 2.1.5.).

No ponto 2.2. descrevem-se as fases subsequentes do processo: a aplicação de algoritmos de mineração (que procuram detectar padrões e tendências que possam transmitir conhecimento de interesse); a visualização da informação (que pretende dar ao utilizador uma visão sobre os dados recolhidos que lhe proporcione um melhor adquirir de conhecimentos); e, finalmente, o pós-processamento (em que se avaliam os resultados).

No capítulo 3 e seus subcapítulos descrevem-se as ferramentas de mineração de texto desenvolvidas, as experiências efectuadas utilizando essas ferramentas e disponibilizam-se os resultados alcançados.

No ponto 3.1. apresentam-se os sistemas "Annotator" (sistema que aplica reconhecimento de nomes de entidades e filtragem – ponto 3.1.1.), "Corparator" (sistema de selecção de atributos por comparação entre corpus – ponto 3.1.2.) e "BDClassifier" (ferramenta de recolha de documentos de interesse baseada em cálculos de similaridade e pesagem – ponto 3.1.3.), estes dois últimos aplicando atomização, redução à forma canónica, remoção de palavras comuns e indexação de documentos.

No ponto 3.2. foca-se a parte experimental, cujo objectivo foi a procura de um modelo de representação para um corpus que garantisse melhores resultados em recolha de documentos "novos" de interesse, no contexto da utilização das ferramentas desenvolvidas. É descrita a metodologia utilizada, e apresentam-se e discutem-se os resultados obtidos.

Por último, no capítulo 4, são retiradas conclusões e é revisto o trabalho efectuado.

2. A mineração de texto

A mineração de texto pode ser definida de modo genérico como um processo de busca de conhecimento numa colecção de documentos com a qual um utilizador interage utilizando um conjunto de ferramentas de análise. Na literatura podem ser encontradas várias definições de mineração de texto [2-4] e até várias designações distintas como por exemplo:

- Mineração de dados em texto [5];
- Descoberta de conhecimento em texto [6, 7];
- Análise inteligente de texto [8].

Uma das definições hoje em dia mais aceite é a apresentada por Hearst [9], que descreve o processo de mineração de texto como sendo a descoberta pelo computador de informação nova e previamente desconhecida, através da extracção automática de informação de várias fontes de texto. Tal como acontece com a mineração de dados, a mineração de texto tem como objectivo a extracção de informação útil a partir de fontes de dados, identificando e explorando padrões de interesse. Também entre as arquitecturas dos respectivos sistemas se podem encontrar muitas similaridades. Ambos incluem rotinas de pré-processamento, algoritmos de descoberta de padrões, e, elementos de apresentação (tais como ferramentas de visualização para permitir diferentes vistas sobre os resultados). Adicionalmente, a mineração de texto adopta nos seus processos de busca de conhecimento muitos dos tipos específicos de padrões que resultaram da investigação na área da mineração de dados. Apesar de se poder apontar tanto em comum, existem algumas diferenças fundamentais. Enquanto no caso da mineração de dados as fontes de informação existem de forma estruturada em bases de dados, na mineração de texto as fontes de informação são colecções de documentos e os padrões interessantes são encontrados por entre a informação textual não estruturada presente nesses documentos. Outra diferença importante é a fase de pré-processamento. Enquanto nos sistemas de mineração de dados o pré-processamento passa pela normalização dos dados e seu estabelecimento organizacional em tabelas, nos sistemas de mineração de texto as tarefas de pré-processamento centram-se na identificação e extracção de características representativas nos textos em linguagem natural presentes nos documentos. Esta fase de pré-processamento é responsável por transformar a informação não estruturada presente nos documentos num formato estruturado intermédio mais explícito, o que não é uma preocupação no caso dos sistemas de mineração de dados [10].

As tarefas de pré-processamento e as operações nucleares de mineração são as duas áreas mais críticas para qualquer sistema típico de mineração de texto e englobam tipicamente processos em série dentro de uma visão generalizada da sua arquitectura. Genericamente, as tarefas de pré-processamento envolvem um processamento e filtragem da informação textual, uma análise que pode ser morfológica, sintáctica e/ou semântica, a classificação dos documentos e a extracção de atributos, transformando o corpus inicial num corpus categorizado, indexado e eventualmente com marcação cronológica. Posteriormente, na fase das operações nucleares de mineração, são aplicados métodos de descoberta de padrões e/ou de análise de tendências. A qualidade dos resultados que o utilizador irá obter depende consequentemente da eficiência da implementação de todas estas tarefas e subtarefas no sistema de mineração de texto. Muitas vezes, entre o pré-processamento e as operações nucleares de mineração a colecção de documentos adquire uma representação plana, comprimida ou hierárquica, de modo a apoiar algumas operações nucleares de mineração tais como a navegação em árvore hierárquica.

Os sistemas de mineração de texto que operam em domínios muito específicos, como a medicina, a biologia e a genética, por exemplo, podem beneficiar significativamente de acesso a suportes de conhecimento adequados ao domínio, tanto numa fase de modelação da representação dos documentos – logo a seguir ao pré-processamento – como numa fase de apresentação e navegação. O conhecimento de suporte, na forma de recursos léxicos, terminológicos e/ou ontológicos é frequentemente utilizado para fornecer restrições a, ou informação auxiliar acerca de, conceitos encontrados na colecção de documentos, tornando a mineração de texto semanticamente mais eficiente.

A mineração de texto pode ser vista como compreendendo como actividades principais (figura 1):

- Uma fase de pré-processamento em que a informação textual é transformada de forma a optimizar a aplicação das técnicas de mineração e que inclui:

  - Uma fase de processamento preparatório dos documentos, em que são transformados em formatos mais adequados à aplicação de técnicas de análise de linguagem natural;

  - Uma fase de aplicação de técnicas de processamento da linguagem natural, que envolve genericamente: uma análise lexical, com a atomização (*tokenization*) do texto em elementos constituintes (como palavras ou números); uma análise morfológica, vulgarmente uma normalização das palavras por redução a forma canónica e filtragem; e uma análise sintáctica, em que usualmente ocorre uma marcação parte do discurso (*POS, part-of-speech*) dos textos e uma análise sintáctica profunda completa ou parcial (*full parsing* ou *shallow parsing*);

- Uma fase de selecção de atributos representativos dos documentos segundo um determinado modelo de representação;

- Uma fase de recolha de documentos de interesse (*information retrieval*) em que são reunidos os considerados relevantes para o cenário em questão;

- Uma fase de extracção de informação em que são identificados e extraídos tipos específicos de informação como entidades e relações entre as entidades.

- Uma fase de mineração dos dados resultantes da fase de pré-processamento de forma a procurar padrões e tendências que possam implicar a descoberta de informação.

- Uma fase de visualização dos resultados da fase de mineração através de tabelas, grafos, hiperligações ou outras formas;

- Uma fase de pós-processamento em que os resultados da mineração são inspeccionados, interpretados e avaliados.

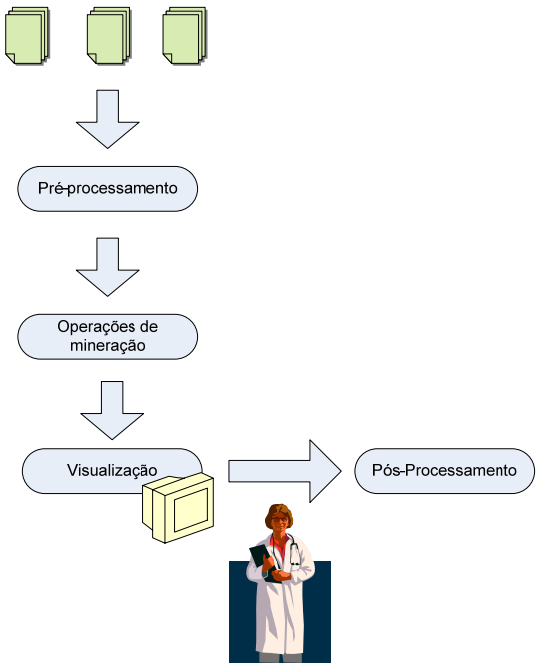

Figura 1 – Actividades principais na mineração de texto.

Este conjunto de actividades envolvidas em todo o processo tornam a mineração de texto uma área verdadeiramente interdisciplinar, sendo necessários conhecimentos da área da mineração de dados tradicional, de linguística, de técnicas de processamento de

linguagem natural, de técnicas de recolha de informação e de técnicas de extracção de informação.

2.1. A fase de pré-processamento

A fase de pré-processamento tem como objectivo a obtenção de uma representação da informação textual não estruturada presente nas colecções de documentos que seja mais adequada à aplicação de algoritmos de mineração. Na Figura 2 é possível ver o seu enquadramento na sequência das actividades principais de mineração.

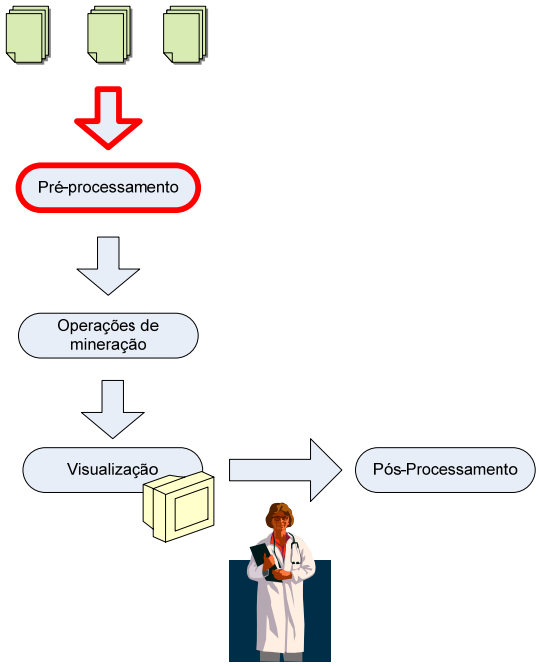

Figura 2 – Enquadramento do pré-processamento nas actividades principais da mineração de texto.

As operações envolvidas nesta fase tratam os elementos presentes num documento em linguagem natural de modo a transformá-lo de uma representação estruturada implícita e irregular numa representação estruturada explícita. Sempre que possível, destas tarefas podem fazer parte quer a extracção quer a aplicação de regras para criação de critérios relativos às datas dos documentos, úteis para posterior análise de tendências.

Um documento é uma entidade abstracta que tem uma variedade de possíveis representações reais. Informalmente, a tarefa do processo de estruturação do documento é pegar na representação em cru e convertê-la para a representação através da qual a essência (isto é, o significado) do documento sobressaia. Para fazer face a este problema é normalmente utilizada uma estratégia de "dividir para conquistar", em que o problema é

dividido num conjunto de pequenas subtarefas, cada uma das quais é resolvida separadamente:

- Processamento preparatório;
- Tarefas de processamento de linguagem natural (NLP) de finalidade geral, que utilizam e produzem atributos linguísticos independentes do domínio;
- Tarefas de classificação dos documentos e de extracção de informação, que são efectuadas em função do problema e lidam portanto directamente com o conhecimento específico de domínio.

Na Figura 3 está representada em árvore a subdivisão em tarefas que envolvem todo o pré-processamento.

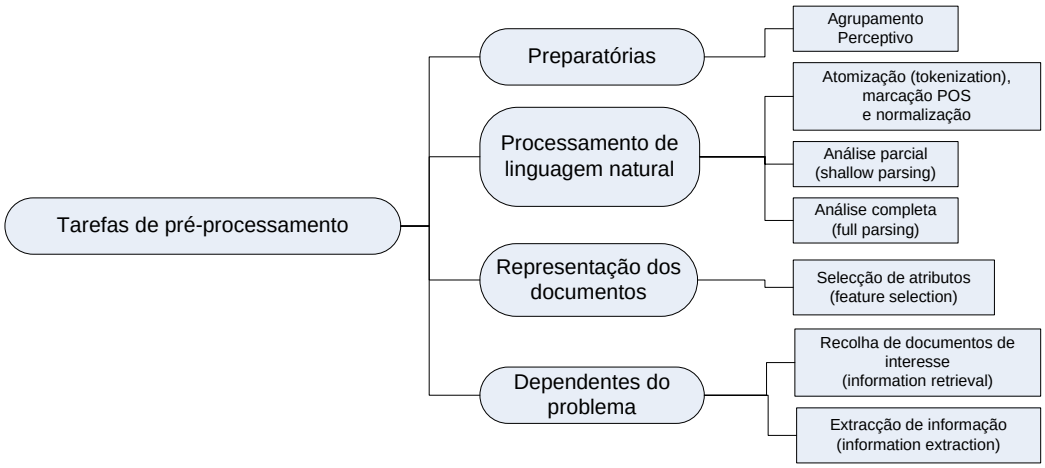

Figura 3 - Representação em árvore das diversas tarefas de pré-processamento

2.1.1. Processamento preparatório

O processamento preparatório converte a representação em cru de um documento numa estrutura mais adequada para processamento linguístico posterior. Na figura 4 é possível verificar como se enquadra em termos de actividades de pré-processamento.

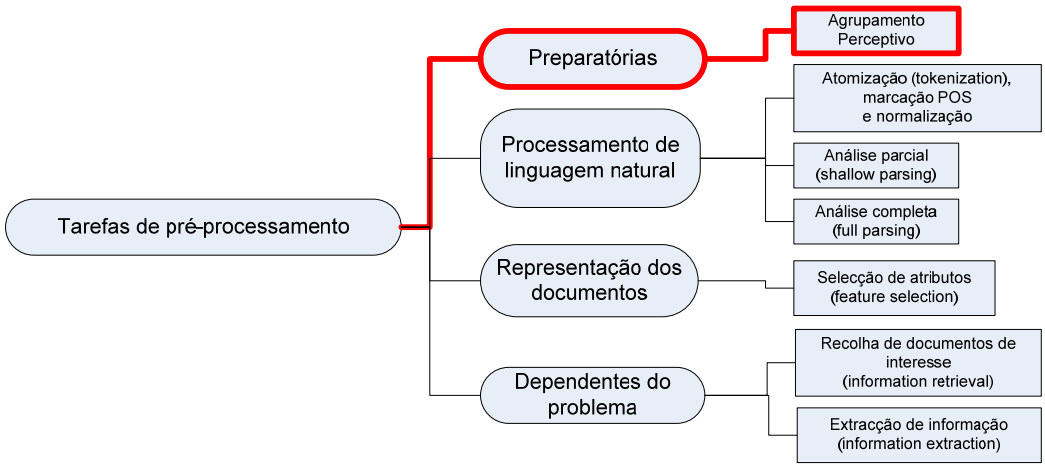

Figura 4 – Enquadramento do processamento preparatório nas tarefas de pré-processamento.

Por exemplo, se a entrada de um sistema for um documento no formato PDF, a tarefa do processamento preparatório poderá ser a conversão num fluxo de texto, a marcação de zonas internas tais como parágrafos, colunas ou tabelas, e a extracção de campos do documento como o autor ou o título. O número de possíveis fontes de documentos, formatos e representações em cru é enorme sendo por vezes necessárias técnicas complexas para conseguir a conversão para um formato conveniente. Exemplos são o reconhecimento óptico de caracteres (OCR), o reconhecimento de voz, ou a conversão de arquivos electrónicos com formatos proprietários. Uma tarefa genérica de processamento preparatório que é frequentemente crítica nas operações de pré-processamento de mineração de texto é o agrupamento perceptivo, que consiste em organizar os elementos constituintes de um documento, como as palavras, parágrafos, títulos, imagens, etc., da forma mais próxima possível à que o ser humano perceptivamente faz. Geralmente este agrupamento resulta numa organização hierárquica em árvore em que os objectos de mais alto nível se ramificam em objectos de nível mais baixo e que reflecte a estrutura conceptual dos documentos. Por exemplo, podemos ter uma organização em que os parágrafos se ramificam em frases, que por sua vez têm folhas, que são as palavras. Um outro exemplo comum de organização perceptiva é a estrutura de um documento HTML, em que a árvore inclui ramos que são o cabeçalho e o corpo, que por sua vez contém tabelas, que contêm formulários, que contêm controlos, e por aí adiante. O modo como a estruturação é concebida, como a organização é pensada e como os objectos são relacionados, depende da aplicação e do formato em causa.

2.1.2. Tarefas de processamento de linguagem natural (NLP)

As tarefas de processamento de linguagem natural (NLP) de finalidade geral processam documentos de texto utilizando o conhecimento genérico acerca da linguagem natural. Dizem-se de finalidade geral no sentido em que as respectivas saídas não são específicas para um determinado problema sendo raramente por si só relevantes para o utilizador final. Destinam-se a um processamento posterior pela aplicação de técnicas de classificação e de extracção de informação, essas sim, dependentes do problema. O conhecimento relacionado com o domínio, porém, muitas vezes pode melhorar o desempenho e é frequentemente utilizado. Na figura 5 pode-se ver como se enquadram as tarefas NLP no pré-processamento.

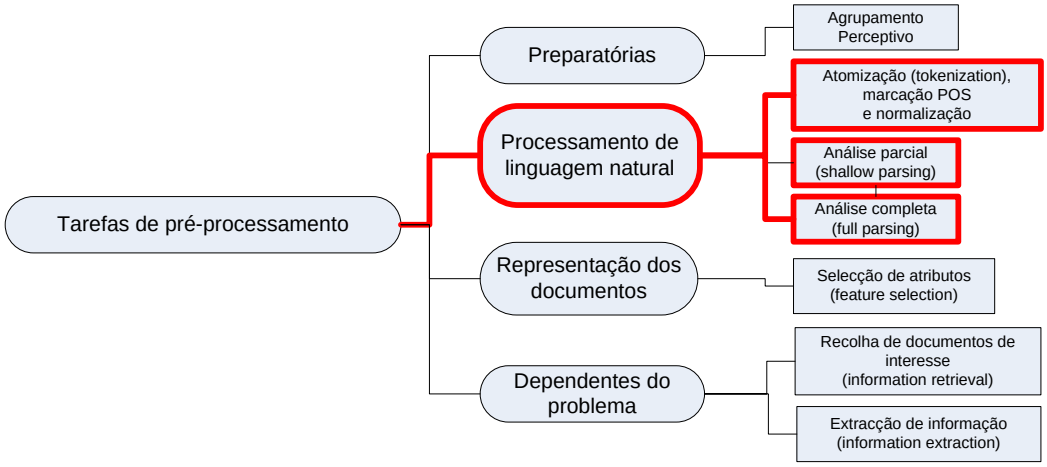

Figura 5 – Enquadramento das tarefas de processamento de linguagem natural.

As tarefas de processamento de linguagem natural (NLP) de finalidade geral processam documentos de texto utilizando o conhecimento genérico acerca da linguagem natural. Dizem-se de finalidade geral no sentido em que as respectivas saídas não são específicas para um determinado problema sendo raramente por si só relevantes para o utilizador final. Destinam-se a um processamento posterior pela aplicação de técnicas de classificação e de extracção de informação, essas sim, dependentes do problema. O conhecimento relacionado com o domínio, porém, muitas vezes pode melhorar o desempenho e é frequentemente utilizado.

É actualmente um parecer ortodoxo que o processamento da linguagem em seres humanos não pode ser separado em componentes independentes. Várias experiências em psicolinguística demonstram claramente que as diferentes fases de análise - fonética, morfológica, sintáctica, semântica, e pragmática - ocorrem simultaneamente e dependem umas das outras. Os algoritmos precisos de processamento de linguagem humana são contudo desconhecidos e embora vários sistemas tentem combinar as etapas num único

processo coerente ainda não foi encontrada uma solução que fosse completa e satisfatória. Assim, a maioria dos sistemas de compreensão de texto empregam a estratégia tradicional de "dividir para conquistar", separando todo o problema em várias subtarefas e resolvendo-as de forma independente. Em particular, é possível chegar muito longe usando apenas linguística, sem conhecimento específico de domínio. Os componentes de processamento de linguagem natural construídos desta forma são valorizados pela sua generalidade. As tarefas que eles são capazes de executar incluem (figura 6):

- A atomização (*tokenization*) e divisão por zonas (*zoning*) – análise lexical;
- Filtragem (remoção de *stop words*) e normalização por redução a forma canónica (*stemming* e *lemmatization*) – análise morfológica - e marcação parte do discurso (*POS tagging*) – análise sintáctica;
- A análise sintáctica superficial ou completa (*full parsing* ou *shallow parsing*).

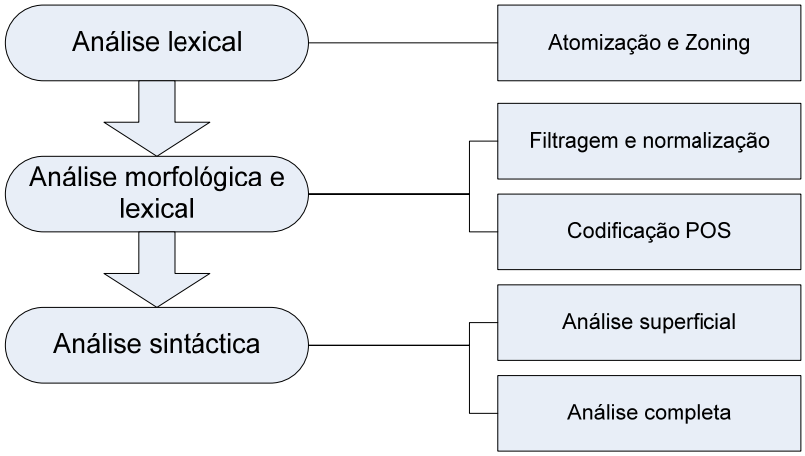

Figura 6 – Tarefas de processamento de linguagem natural do ponto de vista da análise gramatical.

A atomização (*tokenization*) – análise lexical

Antes de processamento mais sofisticado, o fluxo contínuo de caracteres deve ser dividido em constituintes significativos para o problema. Os documentos podem ser divididos em capítulos, secções, parágrafos, frases, palavras, e até sílabas ou fonemas (*zoning*), mas a abordagem mais frequentemente encontrada nos sistemas de mineração de texto envolve uma fragmentação em frases ou palavras, a que se chama atomização.

Um exemplo:

The man, who worked a lot, is going on holiday on Sunday.

[The][man][,][who][worked][a][lot][,][is][going][on][holiday][on] [Sunday][.]

Sendo aparentemente um processo simples, para ser útil e eficiente é preciso ter em conta várias questões, como:

- A presença de abreviaturas;
- A presença de apóstrofes;
- A hifenização;
- Existência de múltiplos formatos de representação;
- A detecção correcta das fronteiras das frases.

O principal desafio na identificação das fronteiras de uma frase num idioma com o inglês, por exemplo, é mesmo conseguir distinguir um período que assinala o fim de uma frase de um período que faz parte de uma unidade atómica (*token*) anterior (como *Mr.* ou *Dr.*). Embora as frases sejam geralmente delimitadas por sinais de pontuação típicos como o ponto final ou o ponto exclamação, por vezes outros símbolos de pontuação são usados como a vírgula, ponto e virgula ou três pontos. A detecção correcta das fronteiras é nuclear caso se queira aplicar posteriormente marcação POS, pois o desempenho depende fortemente da definição das frases. Quanto à presença de abreviaturas no texto, acontece que as palavras nem sempre estão separadas umas das outras por espaços brancos podendo também um ponto significar uma abreviatura e tendo por isso de ser distinguido de um ponto final que é delimitador de uma frase. Este problema agrava-se quando as abreviaturas aparecem no fim da frase, uma vez que nesse caso o ponto possui uma dupla função, interferindo por isso com a detecção das fronteiras de uma frase. No que respeita à existência de apóstrofes, na língua inglesa a contracção de palavras é um procedimento bastante comum. Por exemplo, na frase *IL-10's cytokine systhesis inhibitory activity*, a apóstrofe tem de ser separada da sua palavra âncora (*IL-10*), uma vez que denota uma relação linguística com significado entre duas entidades. É frequente, em domínios como a biologia as entidades terem hifenização e nem sempre é claro se o processo de atomização deve retornar uma ou mais palavras quando é detectada a presença de palavras hifenizadas, especialmente quando os autores não mantêm coerência nas hifenizações. Existem várias entidades que podem aparecer em múltiplos formatos contendo separadores ambíguos. Por exemplo, os números podem aparecer na forma *450678.9* ou *450,678.9*, o mesmo acontecendo com as datas, números de telefone, endereços, etc.

É comum que o módulo de atomização também extraia atributos das unidades atómicas. Estes são geralmente simples funções de categorização descrevendo alguma propriedade superficial da sequência de caracteres que os compõem. Entre esses atributos estão tipos de capitalização, a inclusão de dígitos, pontuação, caracteres especiais, etc.

Normalização: filtragem e transformação em forma canónica – análise morfológica

A normalização tem como objectivo a redução do número de palavras envolvidas nos processos posteriores de mineração, de modo a conseguir um melhor desempenho sem perda significativa de informação útil quanto ao conteúdo dos documentos. Geralmente envolve uma filtragem seguida de uma análise morfológica.

Um método standard de filtragem é a remoção de *stop words*, ou seja, de palavras que usualmente são palavras comuns da linguagem (como conjunções, por exemplo) que comportam pouco ou nenhum significado e que ocorrendo frequentemente ao longo do corpus de documentos têm uma elevada probabilidade de serem estatisticamente irrelevantes. Em algumas situações, existem palavras que mesmo sendo do domínio especifico do corpus de documentos, devido à sua elevada ocorrência podem ser também consideradas *stop words* e portanto ser removidas [11]. Abaixo estão listadas como exemplo 25 palavras semanticamente não selectivas comuns no corpus REUTERS-RCV1.

a an and are as at be by for

from has he in is it its of on that

the to was were will with

O objectivo da análise morfológica é conseguir normalizar as palavras que ocorrem no texto relacionando as diversas variantes de um elemento léxico a uma forma canónica base. As variantes resultam na sua maioria de fenómenos de inflexão (*activat-es*, *activat-ed*) ou derivação (*activat-ion*). Como exemplo, poder-se-ia colocar a hipótese de *activat* ser uma forma canónica base para as três variantes identificadas na frase anterior. Como os fenómenos de inflexão não alteram o significado nuclear da forma base e os de derivação apenas alteram de forma ligeira, reduzir as variantes a uma forma canónica base é um meio simples de unificar a descrição de conteúdo de um documento.

De entre as técnicas mais utilizadas para a análise morfológica, encontram-se a redução a forma canónica por *stemming* e por *lemmatization*, que servem para ilustrar a divisão básica entre as aproximações utilizadas: não necessitando de informação lexical, e recorrendo a informação lexical, respectivamente.

O *stemming* é geralmente implementado com algoritmos baseados em regras que retiram sufixos às palavras, transformando-as em formas canónicas (*stems*) que podem não fazer parte do léxico da linguagem natural em questão, ou seja, podem nem sequer ser palavras propriamente ditas (a forma *activat* é um exemplo). Tem a vantagem de ser de implementação simples e de resultar em algoritmos rápidos, mas tem as desvantagens ser baseado em regras dependentes da linguagem, resultar em formas canónicas que não são palavras e reduzir em muitas situações palavras de significado bastante distinto a uma mesma forma canónica (por exemplo, *army* e *arm* podem eventualmente reduzir-se a uma

mesma forma *arm*). Devido a estas características, o *stemming* é mais utilizado em sistemas de recolha de documentos de interesse em que a exigência de informação semanticamente rica é menor.

Por outro lado, a técnica de *lemmatization*, por análise morfológica com recurso a um léxico, transforma as palavras numa forma canónica (*lemma*) que corresponde à raiz lexical da palavra na linguagem (por exemplo, *processing* transforma-se em *process*). Relativamente ao *stemming* apresenta as vantagens de identificar a raiz propriamente dita das palavras e em geral apresentar menos erros. Tem as desvantagens de necessitar de algoritmos mais complexos e de execução mais lenta. É mais utilizado em sistemas cujo processamento posterior necessita de uma informação semântica mais rica.

Marcação parte do discurso – análise morfológica e sintáctica

Nos passos anteriores, o texto é fragmentado em palavras e frases, que sofrem posteriormente um processo de normalização. Na tentativa de encontrar uma representação cada vez mais estruturada da informação, o passo seguinte é atribuir características morfológicas e sintácticas às palavras ou frases normalizadas, adicionando assim pistas que virão a ser úteis para uma posterior aferição das entidades envolvidas e sobre as relações entre elas. Ou seja, a anotação POS adiciona informação que não se encontra explicita nos documentos, aumentando a sua utilidade para os passos posteriores de mineração. A informação atribuída às palavras é fundamental, por exemplo, na resolução de problemas de ambiguidade aquando de análise posterior, uma vez que adiciona ao texto restrições lexicais e contextuais localizadas.

Em gramática, uma categoria lexical (ou parte do discurso), é uma categoria linguística de palavras que é definida pelo comportamento morfológico ou sintáctico particular, sendo as mais comuns os substantivos, verbos, entre outros. A marcação parte do discurso é a anotação de palavras por atribuição de categorias lexicais.

As categorias principais são as seguintes:

- Nome (*Noun*) – refere-se a entidades, como pessoas, coisas ou ideias;

  *I would like to make a **contribution** to the research*

- Adjectivo (*Adjective*) – descreve as propriedades de nomes ou pronomes;

  *I am **confident** in my ability*

- Verbo (*Verb*) – descreve acções, actividades e estados;

  *in my dissertation I **analysed***

- Advérbio (*Adverb*) – descreve um verbo, um adjectivo ou outro advérbio;

*were **significantly** associated*

- Pronome (*Pronoun*) – palavra que pode tomar o lugar de um nome;

  ***it** was isolated from kidney*

- Determinante (*Determiner*) – descreve a referência particular a um nome;

  ***these** samples were sequenced*

- Proposição (*Preposition*) – expressa relações espaciais ou de tempo.

  *experiences made **in** the laboratory*

O processo de marcação segue um esquema de anotação (*annotation schema*) constituído por:

- Um conjunto de marcadores (*TagSet*) – inventário de marcadores disponíveis;
- Recomendações de anotação (*annotation guidelines*) – informam os anotadores sobre como o conjunto de marcadores deve ser aplicado, garantindo uma consistência entre a anotação efectuada por agentes distintos.

Um exemplo de conjunto de marcadores é o *Penn*, utilizado para o *Penn Treebank* [12] e que é constituído por 45 marcas distintas. Três delas são dadas como exemplo a seguir, na figura 7:

IN	Preposition or conjunction subordinating	astride among uppon whether out inside pro despite on by through out below within for towards near behind atop around if like until below next into if beside ...
JJ	Adjective or numeral, ordinal	third ill-mannered pre-war regrettable oiled calamitous first separable ectoplasmic battery-powered participatory fourth still-to-be-named multilingual multidisciplinary ...
JJR	Adjective, comparative	bleaker braver breezier briefer brighter brisker broader bumper busier calmer cheaper choosier cleaner clearer closer colder commoner costlier cozier creamier crunchier cutter

Figura 7 – Exemplo de marcas POS existentes no Penn Treebank [13, 14].

De forma geral, um processo de marcação POS requer:

3. Uma fase de treino, em que um corpus anotado sintacticamente é processado por um algoritmo de aprendizagem;
4. Uma fase em que um algoritmo de marcação processa os textos utilizando os parâmetros aprendidos na fase anterior.

Existem dois métodos básicos de marcação:

- Marcação baseada em regras;
- Marcação baseada em métodos estatísticos.

Na marcação baseada em regras o marcador aprende as regras linguísticas através de um algoritmo de pesquisa que tem acesso a um conjunto de regras contextuais e lexicais e também a um corpus anotado manualmente [15]. O algoritmo tem, após o processo inicial de atribuição de marcadores ao texto, uma aprendizagem contínua através da proposta iterativa de regras, cujos resultados são comparados com a anotação do corpus. Desta forma, os erros ainda presentes em cada iteração do processo são identificados e as regras são transformadas melhorando a eficiência do marcador. Concretizando, a uma palavra do texto é inicialmente atribuído um conjunto de marcadores possíveis, consoante o esquema de anotação em causa. Seguidamente, são aplicadas regras que vão resultando na eliminação sucessiva de marcadores atribuídos, até se obter o marcador mais adequado ao contexto em que se enquadra a palavra na frase em questão. Por exemplo, suponhamos a frase *"the can"*. Enquanto a atribuição de um marcador à palavra *"the"* é directa (DT – determinante), à palavra *"can"* pode ser atribuído um marcador NN – nome, ou VB – verbo, devido ao duplo sentido da palavra, pois significa num dado contexto o nome "lata" e noutro uma forma do verbo "poder". Através da aplicação de um algoritmo baseado em regras, seriam inicialmente atribuídos os marcadores NN e VB a "*can*". Posteriormente, por aprendizagem, o marcador VB que não é adequado dado o contexto da frase seria removido restando apenas o marcador correcto NN.

Na marcação baseada em métodos estatísticos, a marcação é feita consoante cálculos de probabilidade sobre diferentes ordenações sequenciais de palavras. Genericamente, um corpus anotado é inicialmente analisado registando-se para cada palavra as frequências de ocorrência de atribuição de marcadores podendo também ter-se em conta as palavras adjacentes das frases em questão. Posteriormente, a anotação é efectuada de acordo com os resultados estatísticos efectuados pelo algoritmo. São utilizadas maioritariamente as aproximações:

- N-gram [16];
- Entropia máxima [17];
- Support Vector Machine (SVM) [18];

Existem aproximações híbridas, que tentam aproveitar o melhor dos dois métodos básicos, como a marcação baseada em transformação, da qual o representante mais conhecido será o *Brill Tagger* [19]. Inicialmente, a cada palavra é associado o marcador de maior frequência de ocorrência no corpus de treino, de seguida são aplicadas regras de transformação sensíveis ao contexto que modificam a marcação utilizando como termo de comparação um "*gold standard*" anotado manualmente, e, por fim, a combinação

aprendida entre a aplicação das regras e do método probabilístico é utilizada na posterior marcação de novo texto.

Até há pouco tempo os corpora anotados provinham quase exclusivamente do domínio da linguagem geral presente em jornais e, por isso, os marcadores eram parametrizados de acordo com este nível de linguagem. Devido a este facto, a transposição para outros domínios cuja linguagem apresenta elevada especificidade, como o domínio biomédico por exemplo, era acompanhada de uma grande perda de desempenho. Assim, o simples facto do treino dos marcadores ser efectuado com um corpus de linguagem do domínio, leva a um grande aumento do desempenho, inclusivamente acima do que o estado da arte prevê [20].

Análise sintáctica superficial (*shallow parsing* ou *partial parsing*) e análise sintáctica completa (*full parsing*)

Com o texto anotado por marcação POS, pode-se avançar para uma análise sintáctica mais aprofundada dos textos, nomeadamente ao nível das frases. Existem dois tipos de aproximações:

- A análise superficial ou parcial (*shallow parsing* ou *partial parsing*);
- A análise completa (*full parsing*).

A análise sintáctica completa tenta construir representações hierárquicas completas em árvore (*parse trees*) das frases, utilizando gramáticas. É sempre efectuada de acordo com uma certa teoria gramatical e existe uma divisão básica deste tipo de análise que é feita consoante são utilizadas gramáticas de constituição (exemplo na figura 8) ou gramáticas de dependência (exemplo na figura 9). As gramáticas de constituição descrevem a estrutura sintáctica em termos de frases construídas recursivamente, ou seja, de sequências de elementos sintacticamente agrupados, e a maioria distingue entre frases substantivas, frases verbais, frases preposicionais, frases adjectivas e orações (*clauses*). Cada frase pode consistir em zero ou mais frases mais pequenas ou palavras, de acordo com as regras da gramática. Além disso, a estrutura sintáctica das frases inclui os papéis das diferentes frases. Assim, uma frase substantiva pode ser rotulada dentro de outra frase como o sujeito da frase, o seu complemento directo, ou o complemento.

Por outro lado, as gramáticas de dependência não reconhecem os constituintes como unidades linguísticas separadas, mas focam as relações directas entre as palavras. Uma análise típica de dependência de uma frase consiste num DAG (*Directed Graph*) etiquetado com palavras como nós e relações específicas (dependências) como arestas. Por exemplo, numa frase típica, os nomes correspondentes ao sujeito e ao complemento directo dependem do verbo principal, um adjectivo depende do substantivo que qualifica, e assim por diante. Geralmente, as frases podem ser recuperadas a partir de uma análise de dependência pois são as componentes ligadas do grafo da frase. Além disso, análises de

dependência pura são muito simples e convenientes de usar por si próprias. As gramáticas de dependência, no entanto, têm problemas com certas construções da linguagem comum, como as conjunções.

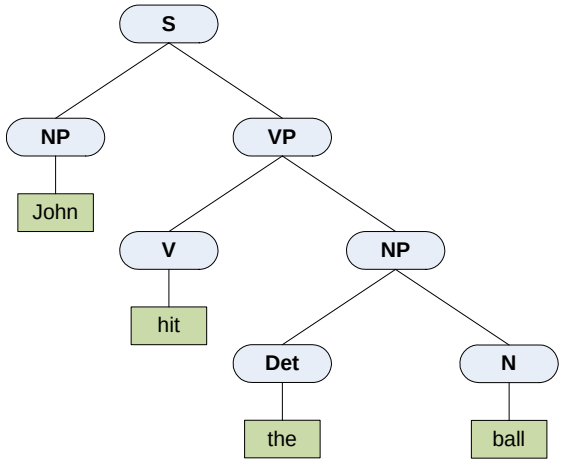

Figura 8 – Exemplo de análise completa com gramática de constituição.

A análise superficial (ou parcial) tenta fornecer informação sobre a estrutura de uma frase sem construir representações hierárquicas completas das frases. É geralmente baseada em regras e técnicas de aprendizagem similares às utilizadas na marcação POS. Estabelece um compromisso em que se obtém velocidade e robustez de processamento sacrificando a profundidade de análise. Em vez de efectuar uma análise completa de toda uma frase, os analisadores superficiais só analisam certas unidades sintácticas que são fáceis e não introduzem ambiguidade. Tipicamente, são geradas frases preposicionais (PP), substantivas (NP) e verbais (VP) pequenas e simples, que ao contrário do que acontece na análise completa, não se podem conter umas às outras. Para conseguir este objectivo é habitual utilizar-se a tarefa designada por *chunking*, que consiste na divisão do texto em segmentos não sobrepostos e não recursivos, ou seja, que não se podem conter uns aos outros (chamados de *chunks*). A seguir tem-se um exemplo de *chunking* efectuado por parte da aplicação *GENIA Tagger*:

[VP Understanding] [NP the monocyte-specific function] [PP of] [NP the]

[NP peri-kappa B factor] [VP may ultimately provide] [NP insight] [PP into]

[NP the different role] [NP monocytes and T-cells] [VP play] [PP in]

[NP HIV pathogenesis] . [NP # # #] [NP This] [VP is] [NP a sample] .

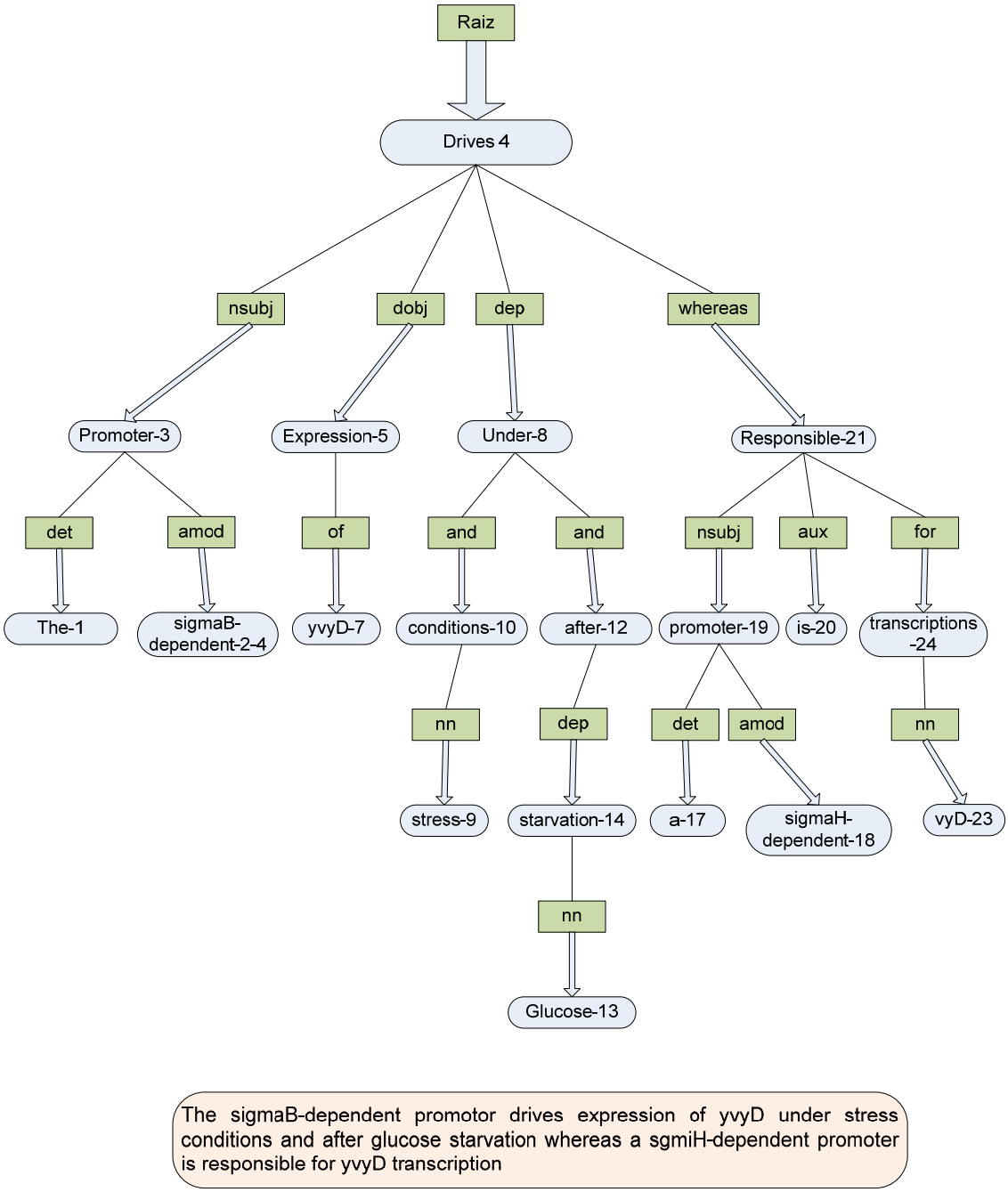

Figura 9 - Exemplo de análise completa com gramática de dependência (Stanford Lexicalized Parser), mostrando palavras (elipses) assinaladas com as suas posições (números ligados às palavras), dependências (setas apontando da cabeça de uma dependência para a palavra dependente), tipos de dependência (rectângulos) e a cabeça da frase (raiz).

Comparativamente, poder-se-á resumir as diferenças entre a análise sintáctica completa e superficial do seguinte modo:

- A análise sintáctica completa resulta numa representação em árvore de toda uma frase, constituída por todos os elementos individuais, enquanto a análise parcial resulta em construções sintácticas simples não recursivas;
- A análise sintáctica completa é muito sensível a ruído (por exemplo, erros tipográficos e de atomização) enquanto a parcial não, pois os elementos geradores de ruído ficam geralmente de fora dos *chunks* criados pelo processo de *chunking*, não causando subsequentemente danos de maior. Numa análise completa, esses elementos são sempre tidos em conta;
- A análise sintáctica parcial introduz uma muito menor ambiguidade, pois existem muito menos alternativas para a divisão de uma frase por *chunking*, do que para a representação numa árvore.

Para efeitos de extracção da informação, a análise superficial é normalmente suficiente e, por isso, preferível à análise completa devido a resultar numa maior velocidade e robustez.

2.1.3. Representação dos documentos

Com o objectivo de optimizar o desempenho dos módulos que executam tarefas de classificação ou agrupamento, é usual tentar obter uma representação para os documentos que seja um subconjunto simplificado de atributos. Na figura 10 encontra-se ilustrado o enquadramento desta actividade na fase de pré-processamento.

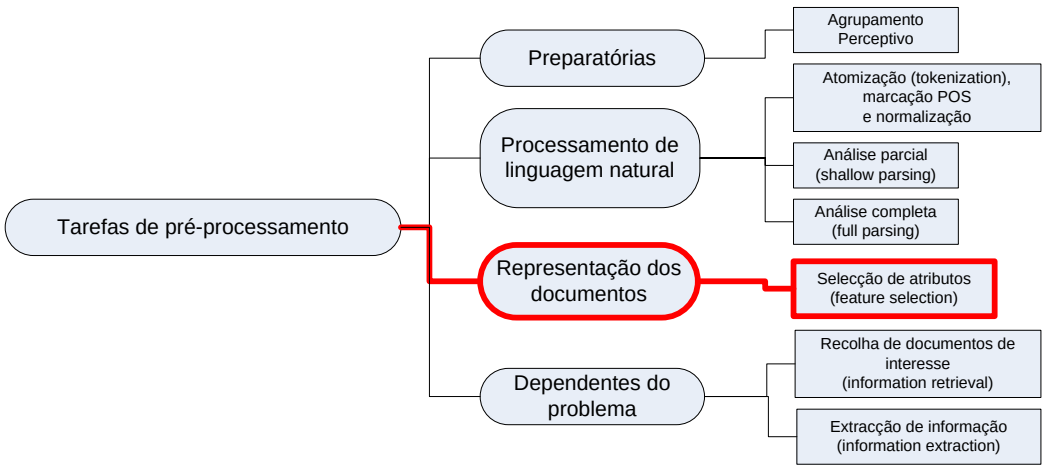

Figura 10 – Enquadramento da actividade de escolha de representação dos documentos.

O conjunto de atributos que representam um documento é designado por modelo de representação e para os seleccionar é geralmente necessário estabelecer um compromisso entre dois objectivos:

- A obtenção duma calibração correcta do volume e do nível semântico dos atributos do modelo que leve a uma melhor captação do significado do documento, o que resulta numa tendência a seleccionar ou extrair relativamente mais atributos para representar os documentos;

- A identificação de atributos de um modo que seja computacionalmente eficiente e que se torne prático para a descoberta de padrões, o que coloca a ênfase na optimização de conjuntos de atributos representativos. Tal optimização é por vezes suportada pela validação, normalização ou referência dos atributos segundo vocabulários controlados ou fontes externas de conhecimento tais como dicionários, ontologias ou bases de dados, resultando em conjuntos mais reduzidos e de maior significado semântico.

Apesar de existirem potencialmente muitos tipos de atributos que podem eventualmente ser utilizados para representar documentos, os mais utilizados são:

- Palavras - são o nível básico de riqueza semântica de um documento e embora seja possível que a representação de um documento contenha todas as suas palavras é mais habitual exibirem uma optimização e consistirem de subconjuntos filtrados de atributos, por processos de remoção de palavras de paragem (*stopwords*), caracteres simbólicos e números sem interesse, de modo a evitar que existam representações com dimensões desnecessariamente vastas;

- Termos - são palavras ou frases, seleccionadas directamente a partir do corpo do documento nativo por metodologias de extracção, resultando normalmente numa representação do documento de menor dimensão e mais rica semanticamente do que a representação por palavras. Muitos dos métodos de extracção convertem o texto numa série de termos normalizados e anotados, sendo por vezes utilizado no processo um léxico externo, que fornece um vocabulário controlado. A selecção dos atributos que representarão um documento é feita pelas metodologias usuais de extracção gerando e filtrando uma lista abreviada dos termos candidatos a mais significativos a partir do conjunto na forma canónica;

- Conceitos - são atributos gerados para um documento através de metodologias de classificação que podem ser manuais, estatísticas, baseadas em regras, ou híbridas. Hoje em dia é habitual extraí-los dos documentos por rotinas de pré-processamento que identificam palavras, expressões, frases ou unidades sintácticas maiores, que são depois relacionadas com identificadores conceptuais específicos. Para os métodos de categorização baseados em regras e manuais, a referência e validação de atributos ao nível do conceito envolve tipicamente a interacção com um "*gold*

standard", tal como uma ontologia, um léxico ou uma hierarquia formal de conceitos. Ao contrário do que acontece com a representação de documentos ao nível da palavra ou termos, ao nível do conceito os atributos podem consistir em palavras que não se encontram no documento nativo.

Em termos comparativos, verifica-se o seguinte:

- Os termos e os conceitos são os que comportam um maior valor semântico existindo diversas vantagens em utilizá-los para a representação de documentos. No que concerne às dimensões dos conjuntos de atributos resultantes, as representações por conceitos e por termos exibem uma eficiência aproximada e de forma geral melhor que as representações por palavras;

- As representações por termos podem muitas vezes ser geradas automaticamente de forma mais simples a partir das fontes de texto originais, através de várias técnicas de extracção de termos, do que as representações por conceitos, que frequentemente implicam em alguma fase uma interacção humana;

- As representações por conceitos, apresentam-se como sendo muito melhores do que qualquer uma das outras no que respeita a lidar com problemas de:

    - Polissémica - palavras que possuem significados distintos consoante os contextos em que são utilizadas;

    - Sinonímia - palavras distintas na forma mas com o mesmo significado;

    - Hiponimia e hiperonimia- termos cujo sentido está incluído na significação de um termo mais abrangente e termos cuja significação inclui o sentido de um ou de diversos termos, respectivamente – um exemplo possível da relação hipónimo/hiperónimo será "legume é hiperónimo de cenoura, logo cenoura é hipónimo de legume";

- As representações por conceitos podem ser processadas para suportar hierarquias sofisticadas e fornecem indiscutivelmente as melhores representações para retratar conceptualmente o domínio em causa, com a ajuda de ontologias e bases de conhecimento;

- As representações por conceitos têm as desvantagens de necessitar de uma maior complexidade das operações de pré-processamento que implementam as heurísticas requeridas para extrair e validar os atributos segundo os tipos de conceitos envolvidos e a dependência do domínio em causa.

Modelo vectorial

A maioria dos estudos na área da classificação de textos adopta um modelo vectorial de representação, nomeadamente o chamado *Vector Space Model* [21, 22]. Um documento é então representado por um vector que é uma sequência de atributos e de pesos calculados consoante um dado critério. O modelo mais simples de representação vectorial, chamado vulgarmente de *bag-of-words*, utiliza todas as palavras de um documento como atributos resultando num espaço cuja dimensão é igual ao número total de palavras diferentes que constituem todo o conjunto de documentos. Esta aproximação não tem em conta a ordem das palavras nos documentos, donde, por exemplo, "Mary is quicker than John" terá uma representação idêntica a "John is quicker than Mary". Contudo, parece intuitivo concluir que documentos que tenham idênticas representações serão muito similares entre si.

As vantagens desta representação são [23, 24]:

- Fácil aplicação a textos provenientes de diversas fontes;
- Baixo requerimento de processamento, especialmente para colecções grandes e dinâmicas;
- Possibilidade de ser usado com ou sem recurso à redução à forma canónica;
- Elevada eficiência;
- Facilidade na aplicação a algoritmos de aprendizagem máquina;
- O facto de existirem muitas situações onde a ordem do texto não é importante.

Os métodos de atribuição de pesos aos atributos podem variar. A aproximação mais simples é a binária em que é atribuído um peso 1 a uma característica caso a palavra correspondente pertença ao documento, ou 0 no caso contrário.

Os métodos mais complexos de pesagem têm em conta a frequência das palavras no documento, na categoria e em toda a colecção de documentos. Nestes, a aproximação mais simples é a utilização da frequência de um termo denominada *Term-Frequency*, $TF(w_i,d_j)$, ou seja o número de vezes que a palavra w_i ocorre no documento d_j (exemplo na figura 11). O problema com esta aproximação é que a palavra pode ser muito comum a todos os documentos, e por este facto perde importância para definir o documento. Por exemplo em textos em inglês sobre genética é muito natural que o termo "gene" ocorra num elevado número de documentos, não tendo relevância para a discriminação de um documento em particular.

Termo	TF	Termo	TF	Termo	TF	Termo	TF
ota	9	toxin	4	compound	2	medium	2
effect	6	countries	3	culture	2	mould	2
aspartame	5	guiven	3	days	2	northern	2
exposure	5	preventiv	3	dna	2	phenylalanine	2
humans	5	rate	3	endemic	2	prevent	2
animals	4	toxic	3	food	2	protein	2
include	4	vitro	3	genotoxicity	2	reactive	2
ochratoxine	4	added	2	incidence	2	synthesis	2
peptide	4	africa	2	induce	2	vivo	2
balkan	4	balkan	2	large	2	weeks	2

Figura 11 – Exemplo de termos e respectivos pesos TF num documento.

De forma a atenuar este efeito, utiliza-se a *Document-Frequency* (DF) que se define como sendo o número de documentos em que ocorre um dado termo t, numa medida designada por *Inverse Document Frequency* (IDF) [25], definida como sendo:

$$idf_t = \log \frac{N}{df_t}$$

Em que N é o numero total de documentos na colecção e df_t a DF para o termo. Assim, a IDF de um termo raro é alta, enquanto a de um termo frequente é baixa. Se utilizado no critério de pesagem dos termos, este método permite reduzir o peso das palavras que aparecem frequentemente, ao mesmo tempo que aumenta muito o peso das palavras infrequentes.

Combinando as definições de TF e IDF, obtém-se um critério de pesagem para cada termo em cada documento, a que se chama *TF-IDF weighting* (*Term Frequency – Inverse Document Frequency*) e que atribui a um termo t num documento d o peso:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \log \frac{N}{df_t}$$

Consequentemente, é atribuído um peso a cada termo que é :

- Mais elevado quando t ocorre muitas vezes mas num subconjunto pequeno de documentos (aumentando o grau de discriminação desses documentos);
- Menor quando t ocorre poucas vezes num documento ou muitas vezes em muitos documentos (o que será um sinal de pouca relevância);
- Menor quando ocorre em virtualmente todos os documentos.

A representação *Vector Space Model* neste caso, então, não é mais que uma representação em matriz, na qual cada linha é um documento, cada coluna uma palavra e cada elemento o peso normalizado do termo, calculado pela fórmula w_{td} = TF x IDF [26].

$$\begin{bmatrix} & T_1 & T_2 & \cdots & T_t \\ D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\ D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \cdots & w_{tn} \end{bmatrix}$$

Na matriz acima, T_1,T_2, … T_t são os t termos presentes no corpus e D_1,D_2, … D_n os n documentos.

Selecção de atributos (*feature selection*)

O número de palavras diferentes é elevado mesmo em documentos relativamente pequenos, como por exemplo em resumos (*abstracts*) de artigos científicos. Assim, a dimensão do espaço vectorial para uma colecção vasta pode atingir a ordem das centenas de milhar, e, por outro lado, os vectores representativos dos documentos individuais poderão apresentar centenas ou milhares de componentes de valor diferente de zero. A maior parte das palavras podem ser consideradas irrelevantes para as tarefas subsequentes de pré-processamento, nomeadamente as tarefas de classificação (categorização ou agrupamento), e portanto podem não ser consideradas sem qualquer impacto negativo no desempenho podendo até resultar numa melhoria devido à redução do ruído. Como exemplo de consequência nefasta da existência de atributos ruído (*noise features*) suponhamos que uma palavra rara como *arachnocentric* não nos dá qualquer informação útil sobre uma classe de documentos *china*, mas por mero acaso ocorre em vários dos documentos utilizados como treino do nosso classificador para essa classe. Então, o método de aprendizagem poderá erradamente produzir um classificador que atribui aos documentos que efectivamente se enquadram num domínio em que *arachnocentric* é semanticamente significativo a classe *china.* A tal generalização incorrecta com origem em propriedades acidentais de documentos chama-se *overfitting* e deve obviamente ser precavida.

O processo de remover as palavras que possam ser consideradas irrelevantes é chamado de selecção dos atributos (*feature selection*). A maioria dos sistemas removem pelo menos palavras comuns da linguagem que normalmente não contribuem para a semântica dos documentos e não adicionam qualquer valor (*stopwords*). Muitos sistemas efectuam uma filtragem muito agressiva removendo cerca de 90 a 99 por cento de todos os atributos.

O critério de filtragem é definido consoante a medida de relevância utilizada para os atributos e as aproximações mais comuns são [27]:

- *document frequency* (DF) - a evidência experimental sugere que utilizar apenas 10 por cento das palavras mais frequentes não reduz o desempenho dos classificadores. Isto parece uma contradição com a lei da recolha de documentos de

interesse que dita que os termos com médias e baixas frequências são os que transportam mais informação. Contudo, não existe realmente qualquer contradição uma vez que a vasta maioria das palavras têm uma frequência DF muito baixa, fazendo com que as que são das 10 por cento mais frequentes sejam de facto palavras com baixa ou média frequência;

- *chi-square* – mede a força máxima de dependência entre um atributo e as várias classes. Em estatística, o teste *chi-square* é aplicado para testar a independência de dois eventos, onde A e B são ditos independentes se P(A|B) = P(A)P(B), ou de forma equivalente, se P(A|B) = P(A) e P(B|A) = P(B). No caso da selecção de atributos, os eventos são a ocorrência do termo e a ocorrência da classe. Um valor alto implica que os eventos apresentam uma maior dependência, ou seja, que a ocorrência de um termo implica uma maior probabilidade de ocorrência de uma classe, ou pelo contrário, da sua não ocorrência.

$$x^2_{max}(f) = \max_{c \in C} \frac{|T_r| \cdot (P(f,c) \cdot P(\bar{f},\bar{c}) - P(f,\bar{c}) \cdot P(\bar{f},c))^2}{P(f) \cdot P(\bar{f}) \cdot P(c) \cdot P(\bar{c})}$$

Na fórmula indicada acima, f é o atributo e c é a classe.

- *information gain* (IG) - mede o número de bits de informação obtidos por predição de classes através do conhecimento sobre se um dado atributo pertence ou não a um documento. Para um termo t_j define-se:

$$IG(w) = \sum_{c \in \{c, \bar{c}\}} \sum_{f \in \{w, \bar{w}\}} P(f,c) \cdot \log \frac{P(c|f)}{P(c)}$$

Por exemplo, para o caso de duas classes:

$$IG(t_j) = \sum_{c=1}^{2} p(Lc) \log_2 \frac{1}{p(L_c)} - \sum_{m=0}^{1} p(t_j = m) \sum_{c=1}^{2} p(L_c|t_j =) \log_2 \frac{1}{p(L_c|t_j=m)}$$

Onde $p(L_c)$ é a fracção de documentos com classes L1 e L2 , $p(t_j=1)$ e $P(t_j=0)$ é o número de documentos com e sem o termo j respectivamente, e $P(L_c|t_j=m)$ é a probabilidade condicional das classes L1 e L2 se o termo t_j está contido no documento ou se está em falta. O que se obtém neste caso é uma medida de quão útil é t_j para prever L_1. $IG(t_j)$ pode ser determinado para todos os termos e aqueles que têm um IG baixo podem ser removidos da representação.

As medidas de frequência que tentam calcular a relevância relacionando os atributos e as categorias, tais como a *chi-square* e *information gain* obtiveram experimentalmente como resultado uma diminuição de um factor de 100 da dimensão do espaço de representação dos documentos, sem qualquer perda de qualidade na categorização e até com alguns melhoramentos [28].

Outro modo de redução do número de dimensões é criar um novo conjunto de atributos sintéticos a partir do conjunto original, correspondendo a uma transformação do espaço vectorial original num de muito menor dimensão. A razão para a utilização de atributos sintéticos em vez das palavras que ocorrem no texto em linguagem natural (como faz o método mais simples de filtragem) prende-se com o facto de devido a fenómenos de polissemia, sinonímia e homonímia as palavras poderem não ser os atributos mais indicados. A transformação poderá possivelmente criar representações de documentos que não sofrem dos problemas inerentes a tais propriedades da linguagem natural. A técnica chamada de agrupamento de termos (*term clustering*) tenta resolver o problema da sinonímia agrupando palavras que têm um grau elevado de relacionamento semântico. Estes grupos são posteriormente utilizados como atributos em vez das palavras originais. Como exemplo simples, supondo os conjuntos de termos A e B, resultados possíveis seriam os grupos G1 e G2:

A = { *attendant, minister, government* } ; B = { *employee, manager* }

G1 = { *attendant, employee*} ; G2 = { *minister, government, manager* }

As experiências conduzidas por vários investigadores mostraram um potencial nesta técnica apenas em situações em que a informação de suporte sobre categorias é utilizada na classificação (categorização) [29, 30]. Com a classificação não supervisionada (agrupamento), os resultados obtidos são inferiores [31, 32, 33].

Uma aproximação mais sistemática é a indexação por semântica latente (LSI - *Latent Semantic Indexing*) [34] que através da aplicação da decomposição por valor singular (SVD – *Singular Value Decomposition*) [34] à matriz termo/documento do espaço vectorial composto pelos atributos originais, consegue obter de uma forma optimizada um espaço vectorial de menor dimensão. A matriz termo/documento é decomposta em três matrizes cujas componentes são linearmente independentes e muitas delas raras podendo portanto ser ignoradas, o que resulta num modelo com um menor número de dimensões. Concretizando, sendo A a matriz termo/documento, então:

$$\mathbf{A = USV^T}$$

Onde U é a matriz cujas colunas são os valores próprios de AA^T, S é a matriz cujos elementos da diagonal principal são os valores singulares de A, e V é a matriz cujos elementos são os valores próprios da matriz A^TA. V^T é a transposta de V. Os valores singulares da matriz A encontram-se segundo esta decomposição ordenados de forma decrescente na diagonal principal de S. Seleccionando os k maiores valores consegue-se uma redução da dimensionalidade. Isto equivale a seleccionar as k primeiras linhas de S e V^T e as k primeiras colunas de U. Esta selecção irá resultar numa representação de A de menor dimensão, pois está livre de "dimensões ruído" e que revela uma estrutura de informação que se encontrava latente (escondida), podendo revelar relações semânticas que de outro modo não seriam facilmente detectadas. A seguir é dado um exemplo.

Suponhamos a seguinte matriz termo-documento (figura 12) e verifique-se que os documentos d2 e d3 não partilham qualquer termo:

	d1	d2	d3	d4	d5	d6
cosmonaut	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0
truk	0	0	0	1	0	1

Figura 12 – Matriz termo-documento.

Agora aplique-se LSI, que resulta numa nova matriz reduzida a duas dimensões (figura 13):

	d1	d2	d3	d4	d5	d6
Dim1	-1.62	-0.60	-0.04	-0.97	-0.71	-0.26
Dim2	-0.46	-0.84	-0.30	1.00	0.35	0.65

Figura 13 – Matriz termo-documento da figura 12 reduzida por aplicação de LSI.

Finalmente correlacionem-se os documentos:

Alta correlação, embora d2 e d3 não compartilhem qualquer palavra

	d1	d2	d3	d4	d5	d6
d1	1.00					
d2	0.8	1.00				
d3	0.4	0.9	1.00			
d4	0.5	-0.2	-0.6	1.00		
d5	0.7	0.2	-0.3	0.9	1.00	
d6	0.1	-0.5	-0.9	0.9	0.7	1.00

Figura 14 – Correlação dos documentos d2 e d3 do exemplo, após aplicação de LSI.

Verifica-se que existe uma correlação alta entre os documentos d2 e d3 que se encontrava latente, o que faz sentido pois os termos *cosmonaut, astronaut e moon* podem ser enquadrados numa mesma área contextual, por exemplo "astronáutica".

A aplicação da técnica LSI sofre da desvantagem de ser computacionalmente muito exigente não sendo hoje em dia ainda uma solução muito utilizada em soluções que trabalham corpus de grandes dimensões.

2.1.4. Recolha de documentos de interesse (*information retrieval*)

A recolha de documentos de interesse (*information retrieval - IR*) é o acto de encontrar os documentos que poderão conter a resposta a uma pergunta em vez de encontrar a resposta propriamente dita [35]. Um seu possível enquadramento na fase de pré-processamento está ilustrado na figura 15.

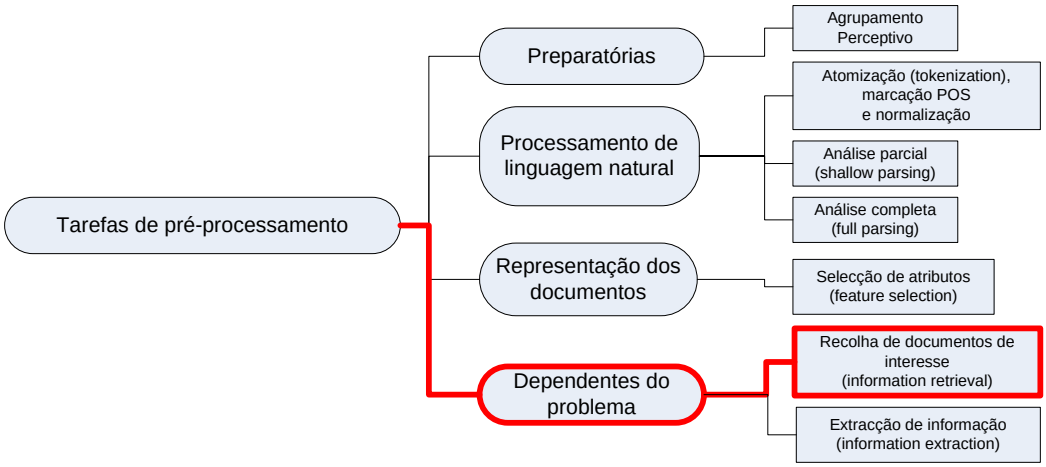

Figura 15 – Enquadramento possível da actividade de recolha de documentos de interesse.

Pode-se hoje em dia afirmar que existem duas abordagens principais:

- Recolha *ad-hoc* de documentos de interesse (*ad-hoc IR*) – em que existe uma consulta efectuada pelo utilizador, e para a qual existem dois modelos:

  - Modelo booleano - *boolean retrieval*;

  - Modelo probabilístico - *term-weighting, ranked* ou *probabilistic retrieval*;

- Recolha por classificação – em que os documentos são classificados e o utilizador pode através das classes aferir da utilidade informativa dos documentos para os seus fins. Os métodos comuns de classificação são:

– Categorização - que é uma classificação supervisionada, em que as classes a atribuir aos documentos se encontra previamente definida;

- Agrupamento – que é uma classificação não supervisionada, em que os documentos são automaticamente agrupados sem o estabelecimento prévio de classes;

Em termos de sistemas do tipo *ad-hoc*, o modelo booleano é um modelo no qual as consultas do utilizador se encontram na forma de expressões booleanas, ou seja, em que os termos se encontram combinados com operadores AND, OR e NOT. Neste modelo, dada por exemplo a consulta "*cats AND dogs*", seriam recolhidos os documentos que contêm ambos os termos e não aqueles que contivessem só um ou nenhum deles. Por outro lado, no modelo probabilístico os documentos são recolhidos consoante consultas que são listas de termos como {*dogs, cats*}, ou frases em linguagem natural como "*I want information on dogs and cats*", por exemplo. Dependendo da consulta são atribuídos probabilisticamente valores aos documentos, que vão ajudar a decidir quais aqueles que melhor a satisfazem. O modelo probabilístico apresenta relativamente ao booleano as vantagens de fornecer informação sobre a frequência de ocorrência dos termos nos documentos em vez de ser apenas armazenado se um termo pertence ou não pertence a um documento, e, além disso, de permitir a recolha de documentos com medidas de similaridade relativamente à consulta (ajudando assim a decidir quais os mais relevantes) em vez de apenas um conjunto solto de documentos sem qualquer informação adicional. Imaginemos que uma dada consulta devolve 1000 documentos, é óbvio que se tornará mais fácil obter aqueles que são mais relevantes se existir um critério de ordenação.

Na recolha por classificação, o utilizador em vez de submeter consultas, pode procurar aferir da utilidade de um documento para si num dado contexto de pesquisa consoante a classe (ou classes) que lhe foi atribuída. A classificação pode ser efectuada por categorização ou por agrupamento, técnicas que serão mais à frente neste capítulo pormenorizadamente abordadas.

Um passo comum a estas abordagens é a indexação dos documentos, que permite um modo rápido de chegar aos documentos a partir de palavras-chave (evitando assim um processamento total dos documentos de cada vez que acontece uma consulta) e também o armazenamento de informação importante (como a *term-frequency*, por exemplo) para cálculos posteriores de similaridade. No caso do possível *pipeline* de mineração que temos estar a seguir nesta dissertação, os termos a utilizar como chaves na indexação seriam os que teriam resultado dos passos de transformação, extracção e selecção de atributos dos documentos anteriormente efectuados, por serem aqueles que melhor os representam. Como a representação vectorial (capítulo 2.1.3) é comummente a utilizada para a recolha de documentos por classificação e para as aproximações probabilísticas à recolha *ad-hoc*, a indexação nestas abordagens é geralmente feita tendo em conta os pesos TF-IDF dos termos. Este tipo de representação e indexação permite simplificar e tornar mais rápido o cálculo de similaridade quer entre documentos, no primeiro caso, quer com consultas (que serão também representadas como vectores), no segundo. A indexação que se tornou standard na área da recolha de documentos de interesse é a chamada indexação *inverted*

índex ou *inverted file* [27]. Nesta abordagem é mantido um dicionário de termos, e para cada termo existe uma lista de registos com ponteiros para os vários documentos (habitualmente uma estrutura de informação mais elaborada) no qual ele ocorre. A cada elemento desta lista, denominada *postings list*, chama-se *posting*. Na figura 16 está um exemplo.

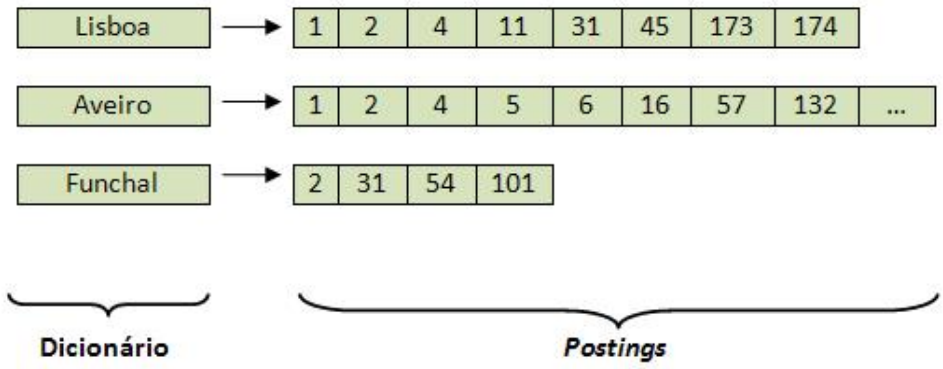

Figura 16 - Exemplo de *inverted índex*. Os termos indexam documentos.

As principais vantagens da utilização do modelo vectorial para o cálculo de similaridades prendem-se com o facto de:

- Permitir uma aproximação matemática simples (cálculo vectorial);
- Considerar tanto a importância local dos termos (TF), como a importância discriminatória global (IDF);
- Possibilitar a ordenação e limitação dos resultados;

A forma mais simples de cálculo de similaridade entre documentos ou entre documentos e a consulta, quando representados segundo o modelo vectorial, é a chamada *cosine similarity* [27]. Esta medida dá-nos o co-seno do ângulo entre os vectores representativos dos intervenientes, utilizando para tal uma normalização do produto interno através do produto entre as normas dos vectores. Ou seja:

$$\mathrm{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}$$

Onde o numerador é o produto interno entre os vectores e o denominador é o produto dos seus comprimentos euclidianos. Na figura 17 é visível graficamente esta relação.

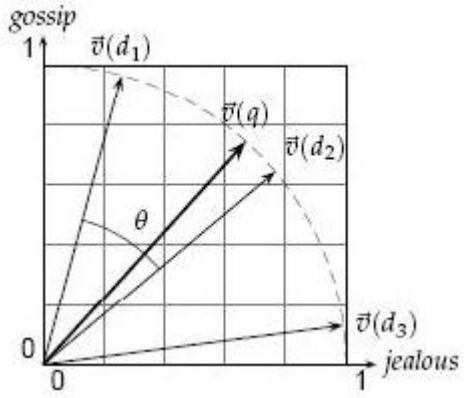

Figura 17 - Exemplo de representação vectorial. O co-seno do ângulo dá uma medida de similaridade[27].

Substituindo na expressão os vectores pelos pesos TF-IDF, virá então:

$$\mathrm{CosSim}(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{iq}^2}}$$

Onde w_{ij} e w_{iq}, são os pesos TF-IDF de cada termo i nos documentos (ou documento e consulta) j e q, respectivamente.

Para a avaliação dos sistemas de recolha de documentos de interesse são utilizadas como métricas a exactidão (*accuracy*) para sistemas multi-classe como a categorização e o agrupamento e as métricas precisão (*precision*) e evocação (*recall*) para sistemas binários como a recolha *ad-hoc*. O sistema é normalmente projectado tendo em conta um compromisso entre a precisão e a evocação pois um valor elevado numa métrica é obtido pela penalização na outra. Para estas métricas de avaliação normalmente recorre-se a quatro unidades. As fórmulas de cálculo das métricas de evocação e precisão dependem do número de documentos correctamente atribuídos a uma dada categoria (a), do número de documentos incorrectamente atribuídos a uma dada categoria (b) e do número de documentos incorrectamente rejeitados para uma dada categoria (c), enquanto a fórmula de cálculo da exactidão além destes valores depende ainda do número de documentos correctamente rejeitados para uma dada categoria (d).

$$recall = \frac{a}{a+c} \qquad precision = \frac{a}{a+b} \qquad accuracy = \frac{a+d}{a+b+c+d}$$

Quando se trata de sistemas de recolha de documentos em que os resultados vêm afectados de medidas de peso que definem o "interesse" relativo para a consulta em causa – *ranked information retrieval* –, uma das medidas mais comummente utilizadas é a *R-precision* [27]. Esta métrica requer a existência de um conjunto de N documentos previamente reconhecidos como "de interesse", a partir do qual se calcula a precisão para os primeiros N documentos que são recolhidos pelo sistema a avaliar. Esta métrica ajusta-

se portanto ao número de resultados definido como conjunto de documentos de interesse a recolher. Um sistema é então à luz desta métrica "perfeito", com R-*precision* de valor 1, quando recolhe exactamente o número de documentos de interesse que deveria em teoria recolher.

Seja então R o número de documentos relevantes para uma dada consulta, e seja r o número de documentos que essa consulta retorna como primeiros R resultados. A métrica R-*precision* é então dada por:

$$\text{R-}precision = r/R \ .$$

Categorização

A tarefa de categorização pode ser descrita como consistindo na classificação de uma porção de informação segundo um conjunto pré-definido de categorias. Quando aplicada à área da gestão documental, fala-se em "categorização de texto", que se pode definir da seguinte forma: dado um conjunto de categorias (assuntos, tópicos) e uma colecção de documentos de texto, a categorização de texto é o processo que consiste em encontrar o tópico (ou tópicos) correcto para cada texto. Uma das definições mais aceites que pode ser encontrada na literatura é a dada por Lewis [36] que a define como "o processo de atribuição de textos a uma ou mais categorias pré – existentes, em vez de os classificar em resposta a uma consulta arbitrária.". Esta última definição realça as diferenças entre categorização de textos e o processo de recolha *ad-hoc* de documentos de interesse, uma vez que este tem por base a divisão dos documentos em dois grupos, os relevantes e os não relevantes para a consulta arbitrária em questão. Além disso, tal como já foi referido, enquanto na recolha *ad-hoc* de documentos é utilizada uma consulta para a selecção dos documentos, no processo de categorização a divisão em classes é feita de forma automática, não como resposta a uma consulta. A categorização também difere do agrupamento de documentos por ser uma classificação supervisionada em contraste com a do agrupamento que é dita não supervisionada (na categorização as classes já se encontram inicialmente pré-definidas, enquanto no agrupamento as classes são aferidas em tempo real pelo algoritmo). Um exemplo esquemático genérico de categorização é ilustrado na figura 18.

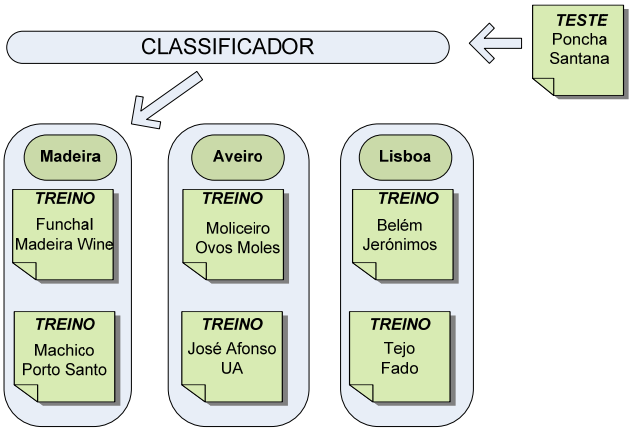

Figura 18 - Exemplo de categorização: o documento de teste é testado e classificado consoante aprendizagem a partir de um corpus de treino.

Existem duas aproximações principais à tarefa de categorização de texto:

- Uma de "engenharia do conhecimento", na qual o conhecimento dos especialistas sobre as categorias é codificado directamente no sistema quer de forma declarativa quer através de regras que impõem os procedimentos de classificação (figura 19);

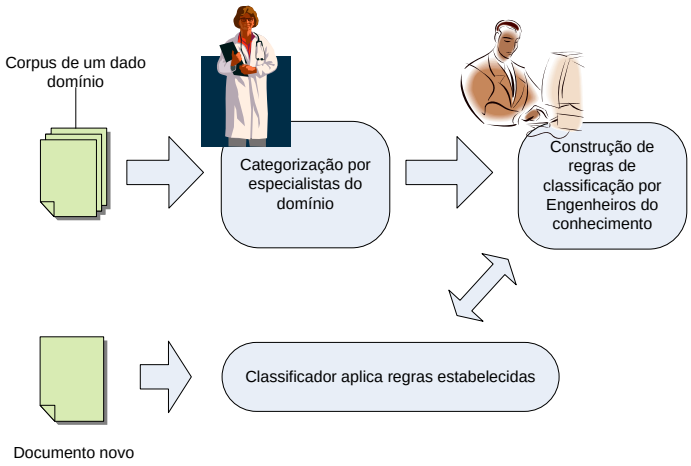

Figura 19 - Sistema de categorização com aproximação "engenharia do conhecimento".

- Uma aproximação "aprendizagem máquina", na qual um processo indutivo genérico constrói um classificador por aprendizagem a partir de um conjunto pré-classificado de amostras (figura 20).

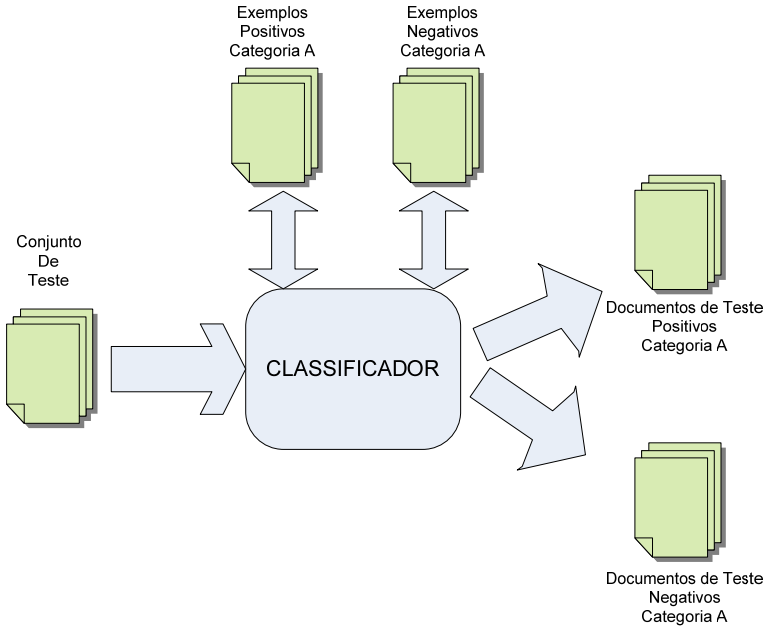

Figura 20 - Sistema de categorização com aproximação "aprendizagem máquina".

Apesar de, na área da gestão documental, os sistemas que utilizam a aproximação "engenharia do conhecimento" apresentarem (por enquanto) melhor desempenho, têm a desvantagem de necessitarem de um grande trabalho especializado e intensivo para a criação e manutenção das regras de codificação do conhecimento. Por este facto a maior fatia do trabalho mais recente na área tem-se focado em sistemas com a aproximação "aprendizagem máquina", que requerem apenas um conjunto de instâncias manualmente classificadas de treino e que têm um custo muito mais reduzido de produção. Adicionalmente neste último tipo de aproximações, pode-se tirar partido da utilização de informação não etiquetada (categorizada), para conseguir melhorar a eficiência da classificação. A tarefa de etiquetar manualmente um número grande de documentos, embora de custo muito menor do que construir manualmente uma base de conhecimento para classificação, é mesmo assim morosa e custosa. Por outro lado, documentos não etiquetados há em abundância e qualquer quantidade pode ser facilmente obtida. Logo, a capacidade de melhorar o desempenho de um classificador através da utilização de um pequeno número de documentos etiquetados aos quais se acrescenta um número grande de documentos não etiquetados é muito útil em qualquer aplicação. Assim, há vantagem em implementar um processo de realimentação automática em que os documentos não etiquetados após sofrerem uma primeira classificação são reutilizados para modelar o classificador. Deste modo, está-se a reduzir a dependência do número de documentos inicial, ao mesmo tempo que se adquirem novos dados sobre a probabilidade conjunta de distribuição de palavras (co-ocorrência). Como exemplo suponhamos que num classificador binário, utilizando numa fase inicial apenas documentos previamente etiquetados, se determina que os documentos contendo a palavra *homework* tendem a pertencer à classe positiva. Os documentos não etiquetados que contenham esta palavra são

então utilizados para redefinir o classificador. O novo classificador determina então que além da palavra *homework*, também a palavra *lecture* tende a aparecer nos novos documentos identificados como pertencentes a classe positiva. Esta co-ocorrência das palavras *homework* e *lecture* ao longo do conjunto de documentos não etiquetados pode leva à construção de um classificador mais perspicaz, que considera *homework* e *lecture* como indicadores de documentos pertencentes à classe positiva nos documentos de treino [37].

De forma genérica podemos considerar que um algoritmo de categorização que utilize informação não etiquetada possui as seguintes fases:

1. O treino do classificador utilizando informação previamente classificada (etiquetada),
2. A atribuição de classificação (etiquetas) aos documentos não classificados,
3. Um novo treino do classificador incluindo os documentos que foram adicionados a cada classe;
4. Uma repetição do processo até ao ponto em que não se verificam mais alterações no classificador.

As duas maneiras mais comuns de introduzir conhecimento com origem em documentos não etiquetados são a maximização de expectativa (EM – *Expectation Maximization*) e o co-treino. Ambas as estratégias apresentaram experimentalmente uma redução significativa (até 60%) na quantidade de informação de treino necessária para produzir o mesmo desempenho do classificador.

De entre as técnicas principais de linguagem máquina aplicadas à categorização de texto estão [38]:

- Os classificadores probabilísticos;
- A regressão bayesiana logística (*Naive Bayes*);
- Os classificadores por árvore de decisão;
- Os classificadores por regras de decisão;
- Os métodos de regressão;
- As redes neuronais;
- Os classificadores baseados em exemplos;
- As SVM (*Support Vector Machines*);

- Os comités de classificação.

Para a maioria destes algoritmos a representação vectorial é adequada não só pelas razões já indicadas anteriormente neste capítulo mas também devido à chamada "hipótese da contiguidade": documentos da mesma classe formam uma região contígua no hiperplano do domínio da função de classificação e regiões correspondentes a classes distintas não se sobrepõem (figura 21).

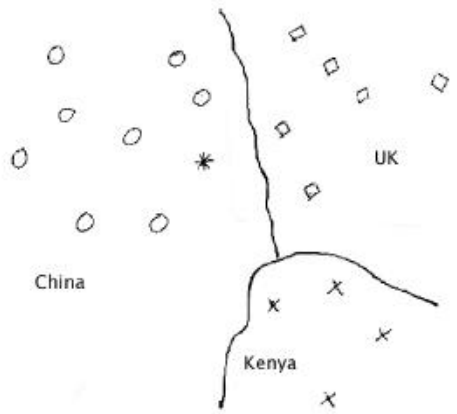

Figura 21 - Espaço vectorial com três regiões, correspondentes a classes distintas. Às linhas chamam-se "fronteiras de decisão" [27].

Como as decisões nestes classificadores se baseiam em cálculos de distância a representação vectorial surge como natural.

De entre as aplicações principais da categorização de texto estão as seguintes:

- Indexação de textos utilizando um vocabulário controlado – se as palavras-chave atribuídas aos documentos na fase de indexação forem vistas como categorias então a indexação de texto é uma instância do problema genérico da categorização e pode ser aproximada pelas técnicas comuns utilizadas em categorização de textos;
- Separação de documentos e filtragem de texto – é a aplicação "clássica" da categorização e consiste em organizar uma dada colecção de documentos em vários "recipientes" virtuais. A actividade de filtragem de textos pode ser vista como uma separação de documentos em apenas dois "recipientes" – o recipiente dos documentos relevantes e o dos irrelevantes;
- Categorização hierárquica de páginas *web* – neste tipo de aplicação, a categorização é utilizada para a classificação automática de páginas *web* segundo catálogos hierárquicos restringindo as pesquisas resultantes das consultas dos utilizadores às páginas que pertencem a um dado tópico. Neste caso, número de documentos que podem pertencer a uma dada categoria é restringido a um dado limiar, impedindo que a sua dimensão se torne excessivamente grande. Sempre que para uma dada categoria o limiar é excedido, isso significa que deverá ser dividida em duas ou

mais subcategorias. Desta forma, o sistema é capaz de criar novas categorias que sejam necessárias e remover as que estejam obsoletas. Na figura 22 tem-se um exemplo retirado do portal *Yahoo*;

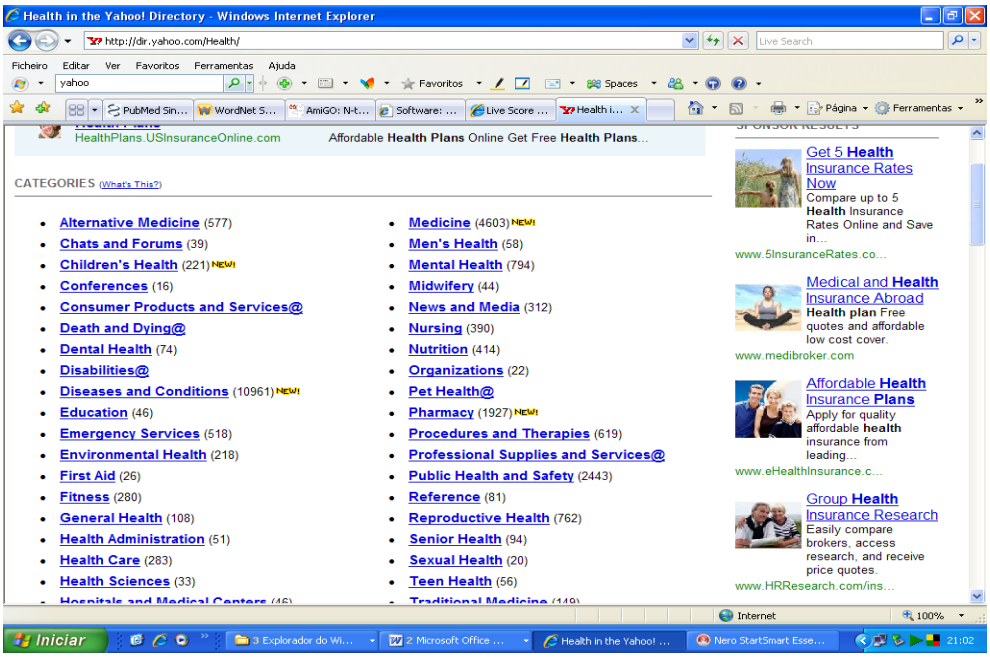

Figura 22 - Exemplo retirado do portal YAHOO.

- Cura de base de dados – é uma das aplicações da categorização de texto mais promissoras e com maior potencial de utilização, pois pode ajudar a aumentar a produtividade através da redução do número de documentos que os curadores têm de rever e seleccionar. Através da categorização é possível com um processo automático seleccionar apenas os documentos que de facto são relevantes.

Agrupamento (*clustering*)

O agrupamento (*clustering*) é um processo não supervisionado de classificação através do qual objectos são organizados em grupos (denominados *clusters*). Enquanto na categorização é fornecido um conjunto de exemplos pré-classificados de treino (ou seja, há supervisão por parte de especialistas que associam previamente documentos a classes) e o sistema aprende as descrições das classes de forma a conseguir classificar documentos, no caso do agrupamento a tarefa é organizar os documentos em grupos (os *clusters*) com significado coerente para o domínio em questão, sem ter como base qualquer informação pré-disponibilizada. Os tópicos associados aos documentos provêm da informação que os documentos disponibilizam por si só através dos seus atributos.

A análise envolvida no agrupamento de documentos consiste na organização de padrões, normalmente representados como medidas vectoriais ou pontos num espaço multidimensional, em grupos, com base em medidas de similaridade (figura 23). Desta

forma, é de esperar que os padrões dentro de um grupo sejam mais similares entre si, do que os padrões pertencentes a um grupo diferente [39].

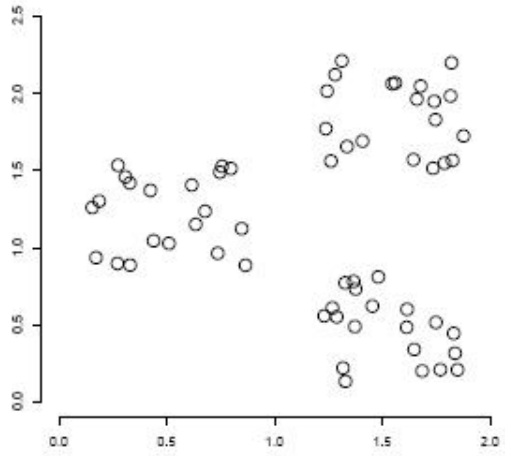

Figura 23 - Exemplo de conjunto de dados com uma estrutura de agrupamento explícita.

Em termos de recolha de documentos de interesse, fala-se em "hipótese do agrupamento" que enuncia que "os documentos de um mesmo grupo têm um comportamento similar no que respeita a relevância relativamente a necessidades de informação" - por outras palavras, os documentos relevantes para uma dada necessidade de informação tendem a ser mais similares entre si do que aqueles que o não são, o que implicará que se um documento de um grupo é relevante, então os outros desse grupo também o deverão ser.

Um processo de agrupamento implica normalmente as seguintes fases:

- Representação do problema - refere-se à definição do número final de classes e do número inicial de padrões bem como à definição do tipo de atributos que estão disponíveis para o algoritmo, sendo o modelo vectorial de representação o mais comummente adoptado;

- Definição de uma medida de similaridade apropriada ao tipo de dados e domínio - normalmente definida por uma distância calculada em função dos atributos utilizados, sendo uma das mais comuns a distância Euclidiana;

- Definição dos grupos de documentos - resulta da aplicação do algoritmo ou algoritmos de agrupamento escolhidos.

Por vezes são efectuados como passos adicionais:

- Uma abstracção dos dados - consiste na extracção de uma representação que seja simples e compacta do conjunto de dados, ou seja, que simplifique uma pós-análise automática ou manual. Um exemplo é a atribuição de títulos

apropriados aos grupos encontrados, que sejam significativos para o utilizador e para o contexto (Figura 24).

- Uma avaliação dos resultados - consiste na avaliação da eficiência do algoritmo de agrupamento, que genericamente pode ser efectuada segundo duas aproximações [10]:

  - Utilizando métricas estatísticas que avaliam a qualidade do agrupamento com base em conexões estatísticas, como a *means square error* e a *silhouette coefficient*;

  - Utilizando comparação com uma dada classificação existente, utilizada como "*gold standard*". Um exemplo de métrica deste tipo é a pureza do agrupamento (*purity*).

Existem diversas variantes de aproximação ao problema do agrupamento:

- Um agrupamento plano produz uma única partição do conjunto de objectos em grupos disjuntos (como na Figura 24);

- Um agrupamento hierárquico resulta numa série em rede de partições (como pode ser visto na figura 25);

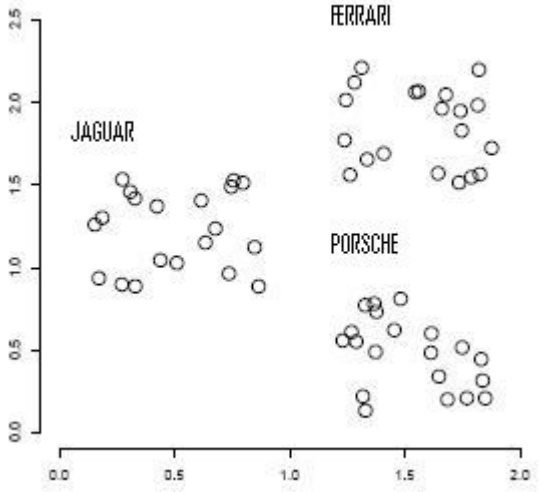

Figura 24 - Conjunto de dados da Figura 23, com atribuição de títulos aos grupos e como exemplo de agrupamento plano.

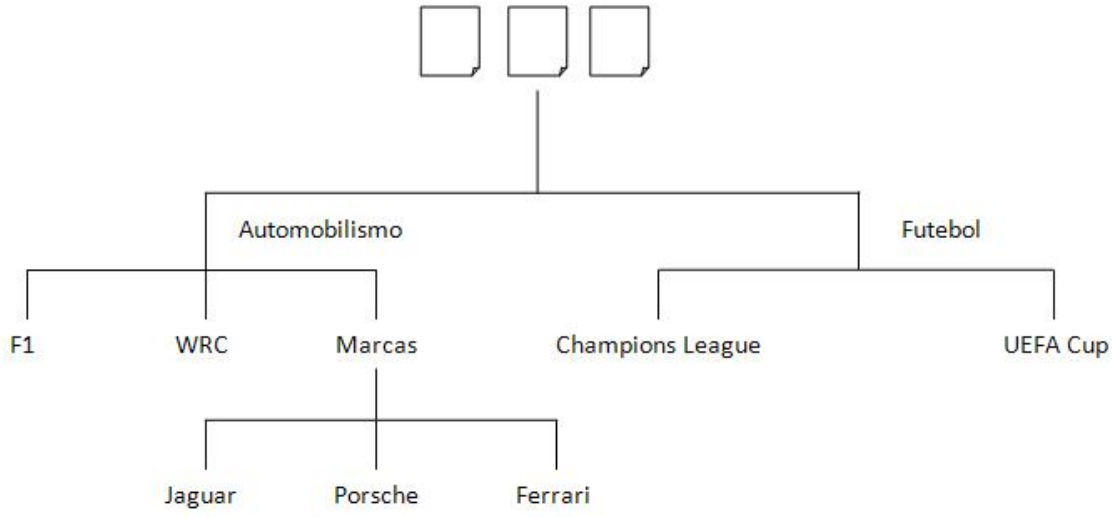

Figura 25 – Exemplo de agrupamento hierárquico.

Cada um dos tipos hierárquico e plano pode por sua vez ser:

- Forte (*hard)* - cada objecto só pode pertencer a exactamente um grupo;
- Suave (*soft* ou *fuzzy)* - os objectos podem pertencer a vários grupos com um dado grau de pertença a cada um deles. Um algoritmo suave pode ser convertido num forte através da simples atribuição de cada documento ao grupo com maior valor de similaridade.

Independentemente do problema de agrupamento em causa, o problema da optimização é computacionalmente muito complexo e em vez de se tentar resolver este problema de forma exacta, o que se faz é tentar conseguir algoritmos que obtenham os melhores resultados possíveis, e que podem ser:

- Algoritmos cumulativos - inicialmente têm-se vários grupos em que cada objecto pertence a um grupo disjunto; posteriormente estes grupos vão sendo fundidos uns com os outros sucessivamente até que um dado critério de paragem seja satisfeito;
- Algoritmos divisivos - inicialmente tem-se um grupo único que contém todos os objectos, que posteriormente se vai sucessivamente dividindo em outros grupos mais pequenos até que um dado critério de paragem seja respeitado.
- Algoritmos de distribuição (*shuffling*) - os objectos são sucessivamente distribuídos em grupos.

Alguns dos algoritmos mais comummente utilizados são:

- O *K-means*, um algoritmo forte, plano e de distribuição, que a partir de k grupos raiz de inicialização (fornecidos externamente ou escolhidos aleatoriamente a partir da representação vectorial) particiona uma colecção de documentos;

- O HAC (*Hierarchical Agglomerative Clustering*), um algoritmo hierárquico e cumulativo, que toma uma distribuição inicial dos documentos em grupos, que vai repetidamente fundindo dois a dois segundo um dado critério.

O agrupamento é muito útil em situações em que existe muito pouca informação pré-disponível sobre os dados e o processo de decisão deve fazer o mínimo de suposições possíveis sobre a natureza dessa informação. Algumas das aplicações principais do agrupamento são:

- Agrupamento de resultados de pesquisa – os resultados de uma pesquisa são agrupados em vez de aparecerem ao utilizador como uma lista que pode ser demasiado longa e sofrer de elevada ambiguidade. Como consequência, obtém-se uma melhoria da precisão (*precision*) das pesquisas. Por exemplo, ao fazer uma pesquisa por "*Jaguar*" é melhor para um utilizador surgirem os resultados agrupados em "*Jaguar Cars*" e "*Cat, Pantera onca*", do que numa lista não estruturada (Figura 26);

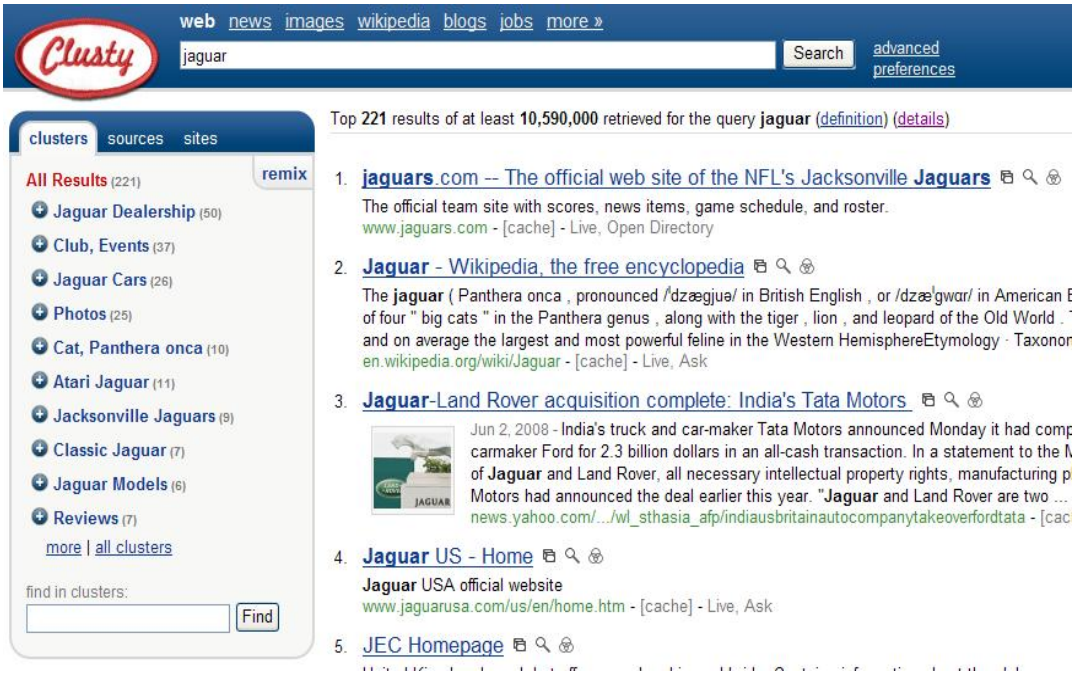

Figura 26 – Exemplo de agrupamento de resultados (coluna na esquerda) em "http://clusty.com/".

- Método de espalhamento/reunião [40, 41] - utiliza o agrupamento como a operação básica de organização. O objectivo é melhorar a eficiência da navegação do utilizador numa colecção de documentos em situações em que a formulação de uma

consulta não é praticável. Durante cada iteração de uma sessão de navegação espalhamento/reunião, um conjunto de documentos é espalhado num dado número de grupos cujas descrições são apresentadas ao utilizador. Com base nas descrições, o utilizador escolhe um ou mais grupos que lhe pareçam relevantes. Os grupos escolhidos são então reunidos numa nova colecção sobre a qual o método pode ser novamente aplicado. O processo é repetido até que um grupo de interesse seja encontrado (Figura 27);

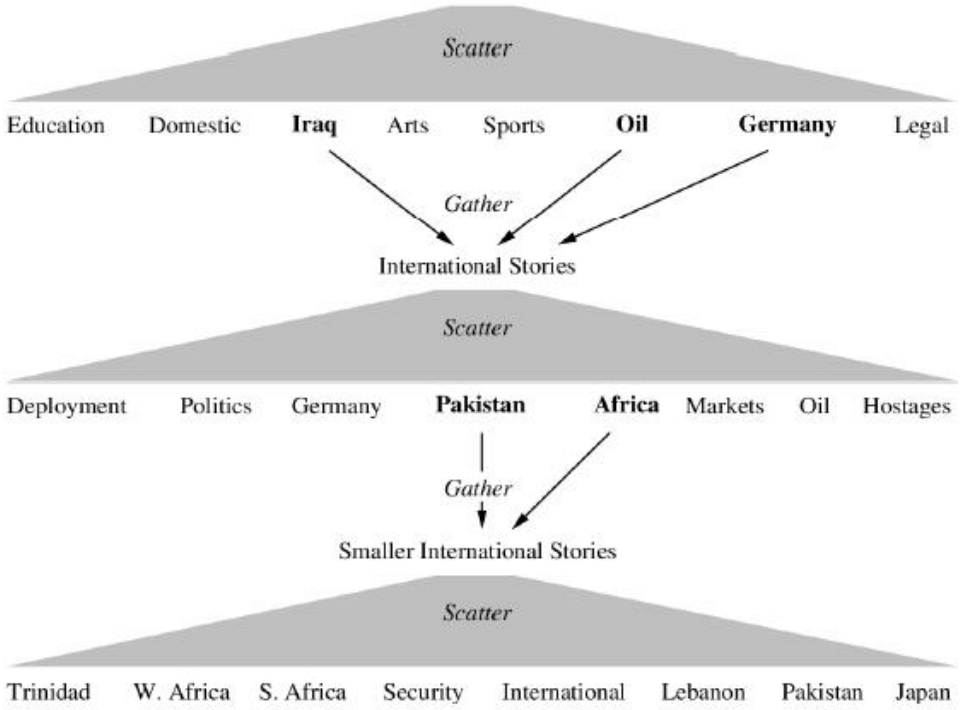

Figura 27 – Exemplo de agrupamento *scatter/gather* [10].

- Agrupamento específico consoante consulta - também são possíveis aproximações que tornam o agrupamento directamente dependente da consulta do utilizador. O agrupamento hierárquico é atraente pois aparenta captar da melhor forma a ideia inerente à hipótese do agrupamento. Os documentos mais similares estarão agrupados nos grupos mais pequenos e coesos, que por sua vez estarão enredados dentro de grupos maiores que contêm documentos com uma menor relação de similaridade. No trabalho descrito por Tombros, Villa e Rijsbergen [42] a hipótese do agrupamento foi testada em várias colecções de documentos e ficou demonstrado que se concretiza para o tipo de agrupamento específico consoante a consulta. Experiências recentes com recolha de documentos baseada em agrupamento específico [43] utilizando modelos de linguagem, demonstraram que este método pode conseguir um desempenho consistente quando estão em causa colecções de documentos de tamanho realista e que uma melhoria é conseguida sem

a necessidade de qualquer tipo de informação do utilizador no que se refere a relevância.

2.1.5. Extracção de informação

A extracção de informação (*information extraction*) consiste no processo de extracção de entidades, atributos das entidades, factos e eventos segundo categorias pré-definidas e na sua representação num modelo (*template*) no qual os campos constituintes (*slots*) são preenchidos com base no que é encontrado no texto. Na figura 28 está ilustrado o enquadramento da extracção de informação nas actividades de pré-processamento.

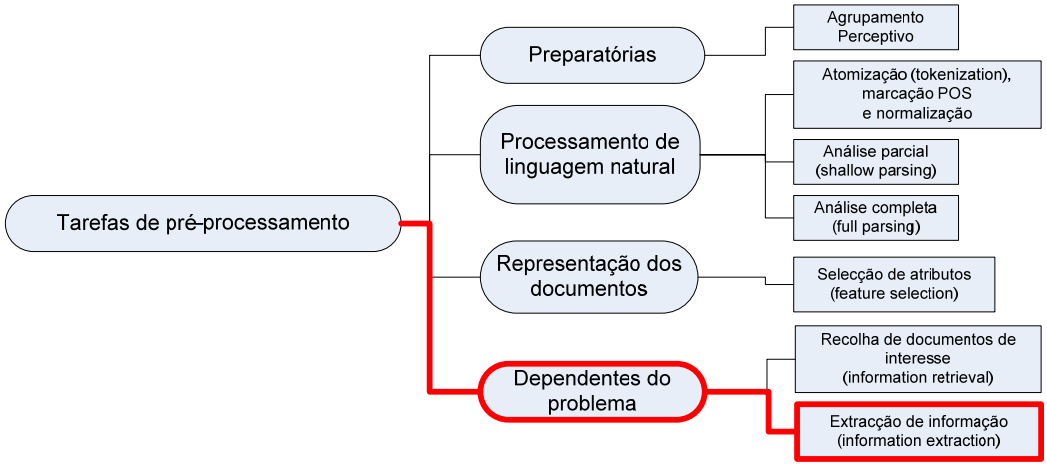

Figura 28 – Enquadramento das actividades de extracção de informação.

A extracção de informação pode ser vista como uma forma conscientemente limitada de compreender um texto. Ou seja, não é feita uma tentativa de compreender o texto no seu todo. O que se faz é definir o tipo de informação semântica a ser extraído do documento.

Existem quatro tipos básicos de elementos que podem presentemente ser extraídos de texto:

- Entidades – As entidades são os blocos construtores de base que podem ser encontrados em documentos de texto. Medicamentos, genes e doenças são exemplos de entidades;
- Atributos – Os atributos são característicos das entidades extraídas, por exemplo a idade de uma pessoa;
- Factos – Os factos são as relações que existem entre entidades. Um exemplo é a fosforilação entre duas proteínas;

- Eventos – Um evento é uma actividade ou ocorrência de interesse na qual entidades participam, como por exemplo um processo metabólico.

Na extracção de informação os documentos são representados por conjuntos de entidades e quadros que descrevem formalmente as relações entre as entidades. Uma forma intuitiva de compreender esta representação é ver um ficheiro XSD (*XML schema definition*) como o modelo, uma instanciação XML que respeite esse XSD como um quadro e os valores e atributos associados às várias etiquetas como as entidades, relações e atributos respectivos. Como exemplo, suponha-se a frase:

"*The new Jaguar XF is a sedan which can cost from $49,200 up to $62,200"*.

Um modelo possível (meramente ilustrativo) para descrever este tipo de informações poderia ser concretizado num ficheiro XSD como o da figura 29 abaixo. Uma concretização desse modelo seria o XML apresentado na figura 30.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Jaguar">
    <xs:annotation>
      <xs:documentation>Information about Jaguar models</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Model"/>
        <xs:element name="Price">
          <xs:complexType>
            <xs:attribute name="minimumPrice"/>
            <xs:attribute name="maximumPrice"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="BodyStyle"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 29 – Exemplo de modelo em XSD para representação de informação sobre automóveis Jaguar.

```
<?xml version="1.0" encoding="UTF-8"?>
<Jaguar xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Jaguar.xsd">
  <Model>XS</Model>
  <Price minimumPrice="$49,200" maximumPrice="$62,200"/>
  <BodyStyle>Sedan</BodyStyle>
</Jaguar>
```

Figura 30 – Exemplo de concretização XML (quadro) do modelo XSD acima.

O tipo mais simples de extracção de informação é a extracção de termos, em que não existem quadros e só existe um tipo de entidade: o "termo".

No coração do processo de extracção de informação está um serviço que recebe os documentos e os processa utilizando um modelo estatístico, um modelo probabilístico, ou uma mistura de ambos. A saída deste serviço é um conjunto de quadros anotados extraídos dos documentos que, por sua vez, preenchem uma tabela na qual cada quadro é uma linha (exemplo na figura 31).

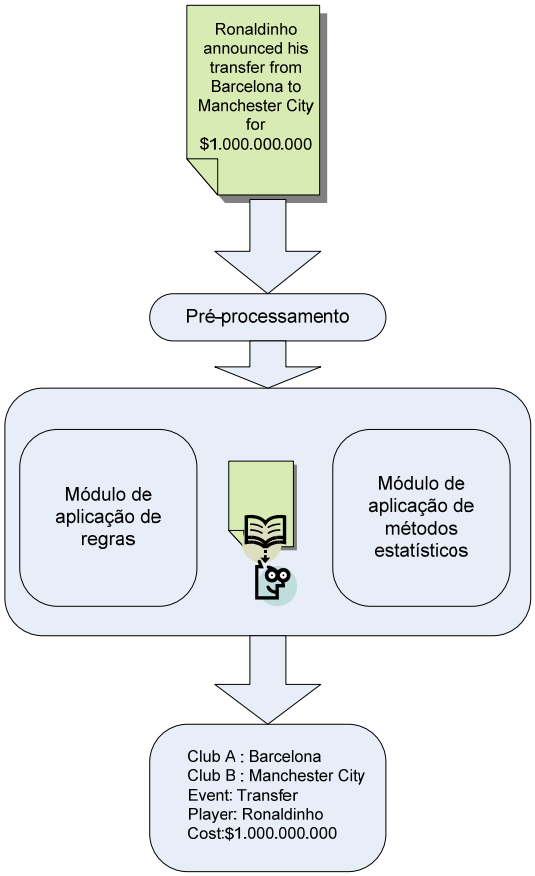

Figura 31 – Esquema de um sistema genérico de extracção de informação.

Em geral, os sistemas de extracção de informação são úteis quando se verificam as seguintes condições:

- A informação a extrair encontra-se explicitamente especificada não sendo necessária qualquer inferência posterior;
- É suficiente um pequeno número de modelos (*templates*) para sumariar as partes relevantes de um documento;
- A informação necessária é expressa de modo relativamente localizado no texto.

Esta é uma tarefa normalmente realizada como suporte a outras tarefas, sendo normalmente utilizada numa sequência de passos de uma aplicação ou processo de mineração de texto. Os resultados obtidos por esta técnica são normalmente guardados em bases de dados para posterior análise por mineração de dados [44], integrados em bases de dados de conhecimento ou apresentados ao utilizador para posterior identificação, associação e comparação dos factos. É importante distinguir entre extracção de informação e recolha de documentos, uma vez que esta técnica tem como objectivo extrair informação do texto sem requerer a leitura do mesmo pelo utilizador. Pelo contrário, como já foi referido, a recolha de documentos tem como objectivo ajudar a encontrar os documentos que melhor cumprem a necessidade do utilizador [24].

As subtarefas envolvidas num processo genérico de extracção de informação são as seguintes:

- Reconhecimento de nomes de entidades (*NER – Named Entity Recognition*) - é a tarefa básica de qualquer sistema de extracção de informação. Nesta fase tenta-se identificar todas as referências de nomes próprios, datas e quantidades no texto. A exactidão (*accuracy*) dos resultados desta fase é habitualmente muito elevada, e os melhores sistemas podem conseguir um valor de até 95% para o ponto de igualdade (*break-even point*) entre precisão (*precision*) e evocação (*recall*). A modificação do domínio dos textos a analisar pode introduzir uma degradação no desempenho da tarefa de reconhecimento de nomes de entidades. Essa degradação irá depender principalmente do nível de generalização utilizado durante o desenvolvimento do motor de reconhecimento de nomes de entidades e da similaridade entre os domínios em causa;

  *"**Fletcher Maddox**, **former Dean of the UCSD Business School**, announced the formation of **La Jolla Genomatics** together with his two sons. **La Jolla Genomatics** will release its product **Geninfo** in **June 1999**. **L.J.G.** is headquartered in the **Maddox family's hometown** of **La Jolla, CA**."*

- Tarefas de obtenção dos elementos dos modelos (TEs - *Template Element Tasks*) – permitem obter os elementos e atributos a utilizar no modelo e são neutrais no que respeita ao domínio ou cenário. Cada elemento do modelo consiste num objecto genérico e alguns atributos que o descrevem, o que permite separar aspectos da extracção que são dependentes do domínio de aspectos independentes do domínio. O que depende do domínio em questão são os atributos das entidades. Exemplos de elementos e atributos a obter são pessoas, organizações e localizações.

  entity {
  ID = 1,
  NAME = "Fletcher Maddox"
  DESCRIPTOR = "Former Dean of USCD Business School"
  CATEGORY = person

}
entity {
ID = 2
NAME = "La Jolla Genomatics"
ALIAS = "LJG"
DESCRIPTOR = ""
CATEGORY = organization
}
entity {
ID = 3
NAME = "La Jolla"
DESCRIPTOR = "the Maddox family hometown"
CATEGORY = location
}

- Tarefa de obtenção dos relacionamentos para os modelos (TR – *Template Relationship Task*) – permite obter os factos para o modelo e expressa um relacionamento genérico entre entidades independente do domínio. Ou seja, tentam-se encontrar as relações que existem entre os elementos que foram extraídos dos textos durante as tarefas de obtenção dos elementos dos modelos. Tal como a ligação dos elementos genéricos dos modelos ao domínio específico em causa é feita através da natureza dos seus atributos, no caso dos relacionamentos são as entidades envolvidas nas relações criadas que contextualizam o domínio.

employee of *(Fletcher Maddox, UCSD Business School)*
employee of *(Fletcher Maddox, La Jolla Genomatics)*
product of *(Geninfo, La Jolla Genomatics)*
location of *(La Jolla, La Jolla Genomatics)*
location of *(CA, La Jolla Genomatics)*

- Tarefa de obtenção de cenários para os modelos (ST - *Scenario Templates*) - tenta expressar entidades e relações específicas do domínio, na forma de eventos.

company-formation-event *{*
PRINCIPAL = "Fletcher Maddox"
DATE = ""
CAPITAL = ""
}
product-release-event *{*
COMPANY = "La Jolla Genomatics"
PRODUCTS = "Geninfo"
DATE = "June 1999"
COST = ""
}

- Resolução de co-referência (CO – *Coreference Task*) – também chamada de resolução de anáforas, captura informação sobre expressões de co-referenciação (por exemplo, pronomes ou outras menções a uma dada

entidade), incluindo aquelas identificadas e etiquetadas nas tarefas de obtenção de elementos e de obtenção de relacionamentos. Esta tarefa baseia-se na relação de identidade (IDENTITY), que é simétrica e transitiva. Cria classes de equivalência (ou cadeias de co-referência) utilizadas para pontuação. São marcados nomes, pronomes e frases com nomes. Por exemplo,

"*David$_1$ came home from school, and saw his$_1$ mother$_2$, Rachel$_2$.*

She$_2$ told him$_1$ that his$_1$ father will be late."

Resulta numa identificação das cadeias de co-referência:

(David$_1$, his$_1$, him$_1$, his$_1$) e (mother$_2$,Rachel$_2$, She$_2$)

Como primeiro passo da etiquetagem (ou marcação) dos documentos a serem explorados por sistemas de mineração de texto, cada documento é processado para se encontrarem (ou seja, extraírem) entidades e relações que possam com um grande grau de probabilidade conter informação significativa para o domínio em causa. As entidades são reconhecidas pelo processo de reconhecimento de nomes de entidades. As relações a que nos referimos no contexto da extracção de informação podem ser factos ou eventos envolvendo certas entidades. Um exemplo de evento seria a entrada de uma empresa num grupo de desenvolvimento de um novo medicamento, enquanto um facto seria, por exemplo, o conhecimento de que um gene provoca uma doença. Os factos são estáticos e normalmente não mudam. Os eventos são mais dinâmicos e geralmente datados. A informação extraída fornece dados mais concisos e precisos para o processo de mineração do que as aproximações baseadas em palavras, tais como por exemplo as utilizadas em categorização, e essa informação tende a representar conceitos e relações que têm um significado mais rico e que se relacionam directamente com o domínio do documento em questão. Consequentemente, os métodos de extracção de informação permitem que se faça posteriormente uma mineração sobre a informação de facto presente no texto e não sobre um conjunto de marcadores ou etiquetas associadas ao documento. O processo de extracção de informação, por outro lado, torna "ilimitado" o número de entidades e relações sobre as quais a mineração pode ser feita, tipicamente na ordem dos milhares ou até milhões, o que ultrapassa largamente o número de etiquetas com que qualquer sistema de categorização automática consegue lidar. Desta forma, as técnicas de pré-processamento que envolvem a extracção de informação tendem a criar modelos de representação mais ricos e flexíveis para os documentos.

2.2. Restantes fases da mineração de texto

No capítulo anterior apresentaram-se os passos de pré-processamento mais comuns num sistema de mineração. O objectivo das tarefas envolvidas é tentar estruturar a informação obtida dos documentos de forma a tornar a aplicação dos algoritmos de mineração propriamente ditos o mais simples e útil possível. Uma vez efectuada a mineração, há que apresentar os resultados ao utilizador, permitindo-lhe visualizar uma representação dos dados que seja informativa e permita adquirir novos conhecimentos. Os dados podem eventualmente sofrer um pós-processamento que se ache adequado. Assim, após o pós-processamento, as restantes tarefas de mineração dividem-se em:

- Operações nucleares de mineração;
- Apresentação e visualização da informação;
- Técnicas de refinamento (pós-processamento).

As operações nucleares de mineração

As operações nucleares de mineração, também referidas como processos de destilação do conhecimento, são o coração de um sistema de mineração. Incluem:

- A descoberta de padrões – como co-ocorrem conceitos presentes no corpus;
- A análise de tendências – qual a tendência temporal de ocorrência de conceitos;
- Os algoritmos de descoberta progressiva do conhecimento – tentam lidar com o problema da actualização da análise em corpus muito dinâmicos.

Enquadram-se no *pipeline* geral da forma ilustrada na figura 32.

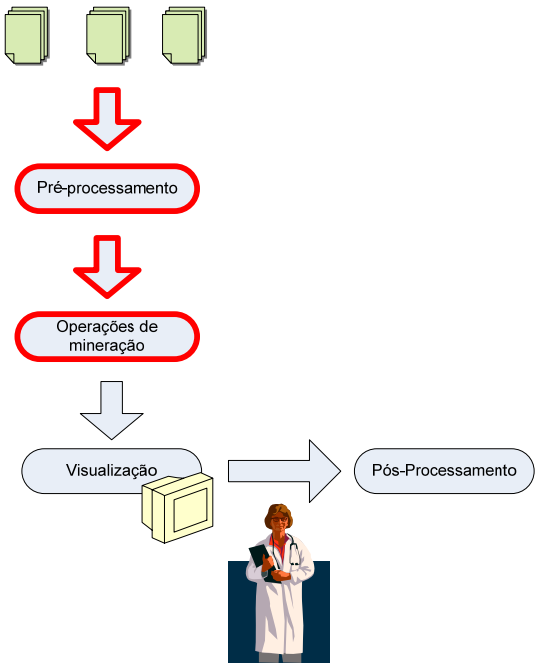

Figura 32 – Enquadramento das operações de mineração.

Os sistemas de mineração de texto utilizam aproximações algorítmicas e heurísticas de forma a considerar:

- Distribuições (e proporções) – distribuição probabilística de conceitos ao longo do corpus e sua proporção relativa – por exemplo o utilizador pode averiguar num corpus noticioso como a ocorrência do conceito "*Iraq*" se relaciona com a ocorrência do conceito "*Al Qaeda*";

- Conjuntos de conceitos frequentes e próximos frequentes (*near-frequent*) – co-ocorrência de conjuntos de conceitos num dado número limiar de documentos, e como se sobrepõem (relacionam de forma não direccionada) sobre o corpus estes conjuntos frequentes, respectivamente – por exemplo a co-ocorrência dos conceitos "*Portugal*", "*Tourism*", "*Algarve*" e sua relação com a co-ocorrência dos conceitos "*British*" e "*Golf*";

- Associações – relações direccionadas entre conceitos ou conjuntos de conceitos, ou seja, como um dado conceito A implicam de alguma forma a ocorrência de um conceito B – por exemplo, analisando os relatórios de vendas de um supermercado, retirar informação como "*75% das transacções que incluem fraldas também incluem toalhetes. 12% de todas as transacções contém ambos os itens.*"

Os métodos de descoberta de padrões da mineração de texto têm como objectivo a descoberta de co-ocorrência de relações entre conceitos, tal como é reflectida pela totalidade do corpus em questão. Esta análise ao nível dos documentos é executada de forma que um utilizador possa descobrir a natureza e as relações entre conceitos reflectidas

pela colecção de documentos como um todo. Por exemplo, pode ser inferida uma potencial relação entre duas proteínas P1 e P2, através de um padrão do tipo:

a) Vários artigos mencionarem a proteína P1 em relação à enzima E1;

b) Alguns artigos descreverem similaridades funcionais entre as enzimas E1 e E2 sem referir nomes de proteínas;

c) Vários artigos relacionarem a enzima E2 com a proteína P2.

Como é bem reflectido neste último exemplo, a informação, em geral, não é fornecida por um só documento, mas sim pela totalidade da colecção.

Outra tarefa comum da mineração de textos é a análise de tendências, que se baseia numa análise do comportamento da distribuição de conceitos através de múltiplos subconjuntos de documentos ao longo do tempo. A análise de tendências procura responder a questões tais como por exemplo, para um domínio jornalístico, "Qual a tendência geral dos tópicos noticiosos entre os períodos X e Y? (com cada período a ser representado por um respectivo subconjunto de documentos) ". Uma outra forma de mineração dependente da marcação temporal dos documentos é a análise de associações efémeras, que se definem como relações directas ou inversas entre as distribuições probabilísticas de conceitos num dado intervalo de tempo. Estas associações são comuns em domínio noticioso em que a ocorrência muito frequente de um dado tópico durante um certo período influencia a emergência (relação directa) ou desaparecimento de outros tópicos (relação inversa). Um exemplo está ilustrado na figura 33.

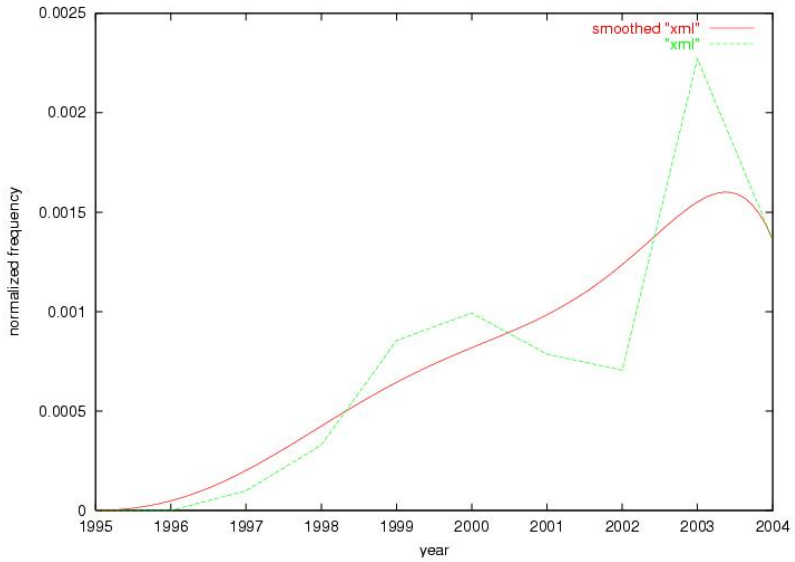

Figura 33 – Gráfico ilustrativo de uma análise de tendências, neste caso segundo a frequência de ocorrência do termo "XML" em documentos de uma livraria digital [45].

Em situações em que a colecção de documentos é muito dinâmica existe a necessidade do utilizador do sistema de mineração possuir um conhecimento devidamente actualizado sobre os padrões e tendências, contando com a informação providenciada pelos novos documentos adicionados. Uma solução óbvia mas que se mostra ineficiente em termos de desempenho é processar novamente a análise sobre todo o corpus modificado. Mais inteligente e útil é aproveitar o conhecimento recorrente das análises efectuadas sobre o corpus (tal como era antes das alterações), como base à qual nova informação pode ser adicionada de forma progressiva (incremental). Ao tipo de algoritmos baseados neste princípio dá-se o nome de algoritmos de descoberta progressiva ou incremental do conhecimento.

A apresentação e visualização da informação

Na mineração de textos, assim como em muitas outras áreas em que existem resultados de teor cientifico a pesquisar e/ou a apresentar a um utilizador, é muito importante a forma como é feita a apresentação e visualização da informação, especialmente em sistemas que têm uma intervenção humana no processo de descoberta. É recomendável que o utilizador tenha disponíveis possibilidades robustas de navegação e que a informação, muitas vezes constituída por padrões densos e de difícil formatação, esteja disponível permitindo uma eficiente interacção e exploração, podendo levar à descoberta de novos conhecimentos. Na figura 34, vê-se como a apresentação e visualização se enquadra em todo o *pipeline*.

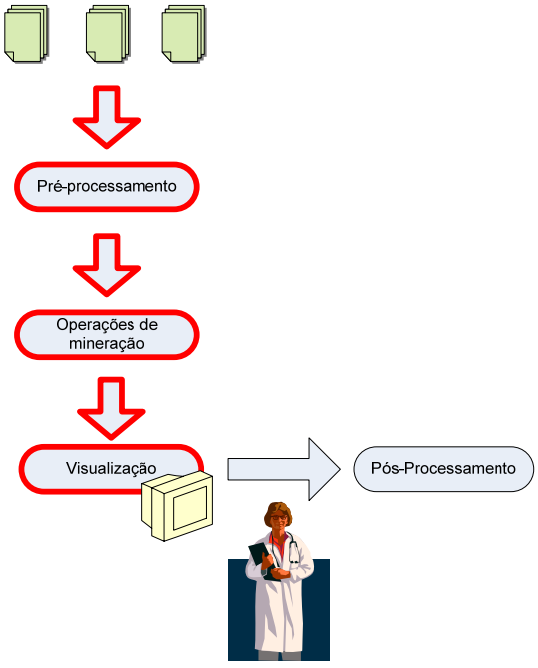

Figura 34 – Enquadramento da fase de visualização.

A visualização gráfica de informação é frequentemente mais intuitiva e compreensível do que seria se fosse apresentada como texto e portanto mais adequada à

mineração de colecções vastas de documentos. Consequentemente, os métodos de visualização quando utilizados em sistemas de mineração de texto, podem ajudar à descoberta e extracção de padrões relevantes de conhecimento. Existem vários tipos de informação que podem permitir uma representação gráfica, tais como aspectos da colecção de documentos, resultados de operações de mineração, relações entre conceitos, ontologias, etc., assim como existem várias aproximações possíveis para a apresentação da informação, das quais são exemplo:

- Apresentação e navegação por distribuições – permite a investigação do conteúdo de um conjunto de documentos ordenando-o segundo a distribuição dos nós de uma dada hierarquia de conceitos. Quando são analisados desta forma e a distribuição é apresentada, o utilizador pode aceder aos documentos específicos de cada subgrupo [10]. A figura 35 mostra um exemplo;

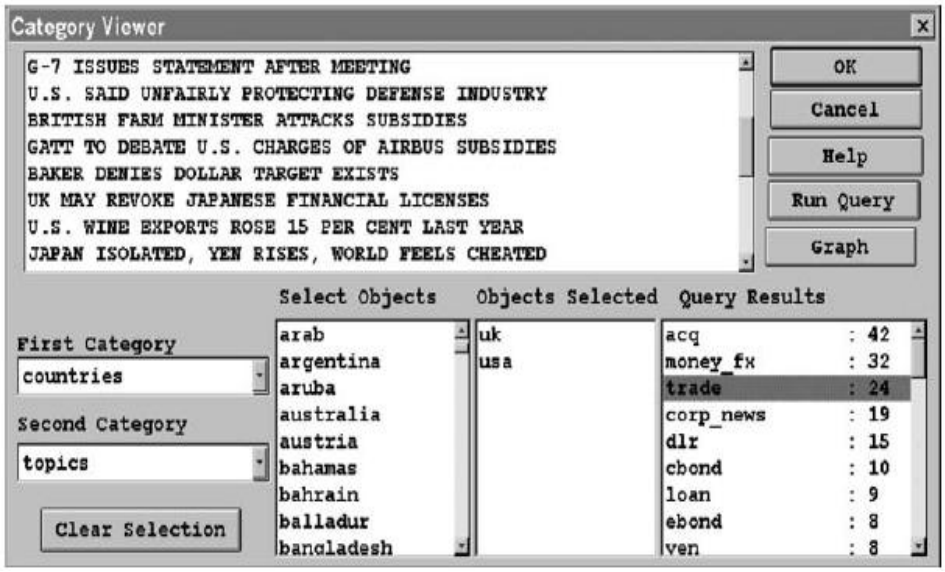

Figura 35 – Navegador de distribuição de conceitos [10].

- Apresentação e exploração de associações – ajudam o utilizador a identificar por entre todas as associações resultantes de um processo de mineração, aquelas que são relevantes podendo levar à descoberta de conhecimento novo [10]. Na figura 26 ilustra-se um sistema deste tipo;

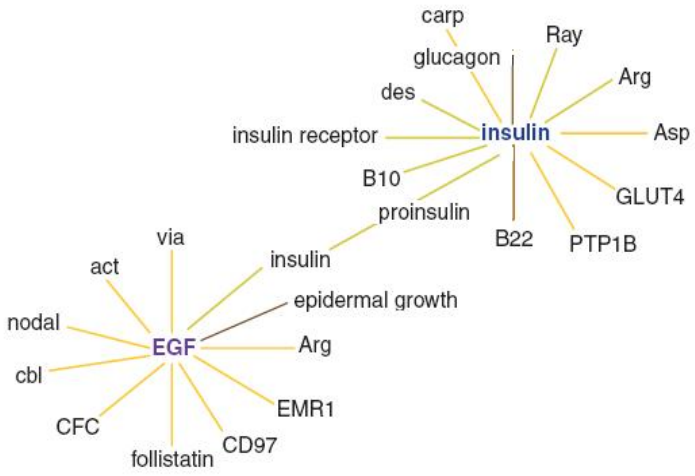

Figura 36 – Grafo de associação que mostra uma relação de co-ocorrência entre frases que referem genes com ferramenta *ClearResearch* [46].

- Navegação e exploração através de hierarquias de conceitos – o utilizador possui a possibilidade de interagir com uma organização hierárquica e com significado coerente dos conceitos, podendo assim explorar interactivamente a informação disponibilizada [10]. Na figura 37 é visível um exemplo;

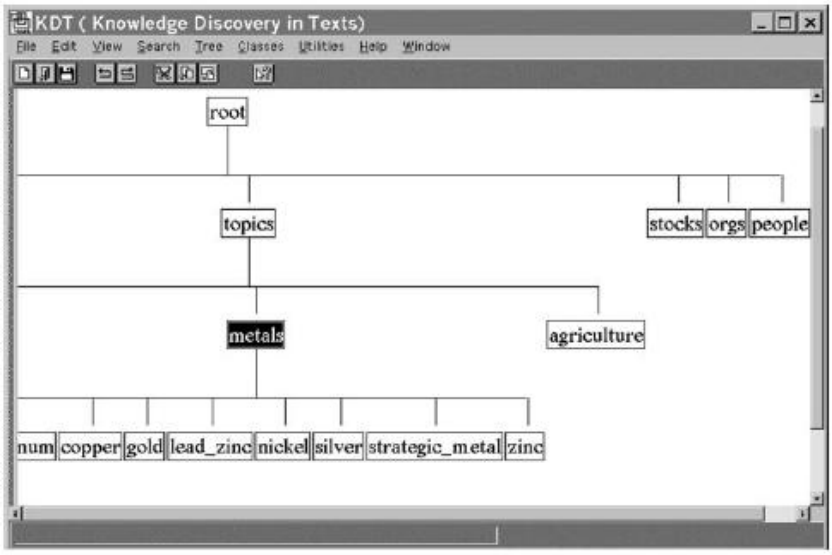

Figura 37 – Interface gráfico simples ilustrativo da exploração de uma hierarquia de conceitos, neste caso para manipulação de uma taxonomia [10].

- Exploração da informação utilizando agrupamento – permitem ao utilizador estabelecer critérios complexos para operações de agrupamento e visualizar os resultados [10]. Pode-se ver um exemplo na figura 38;

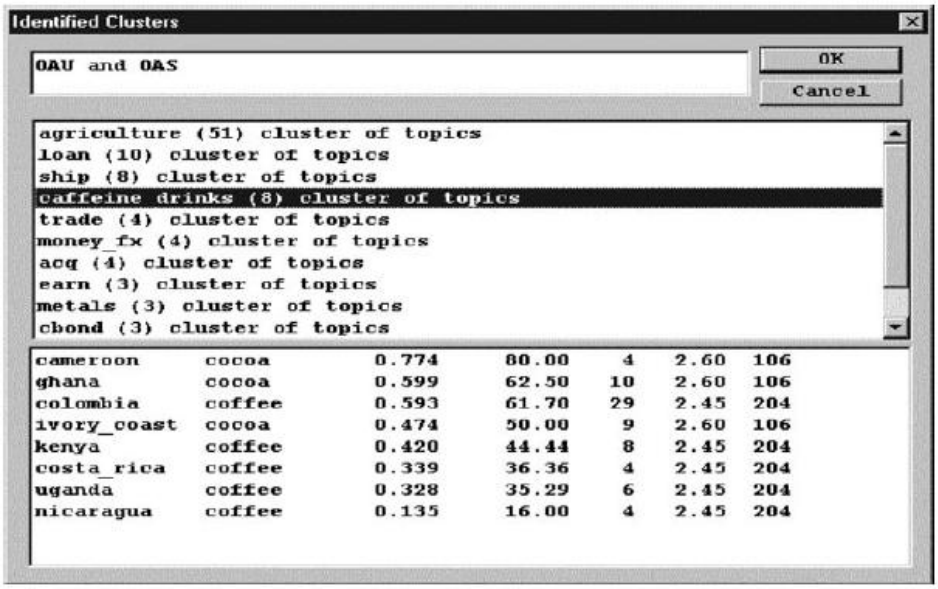

Figura 38 – Associações por agrupamento utilizando uma hierarquia de categorias [10].

O pós-processamento

No final da sequência de etapas constituintes de uma tarefa de mineração de texto (figura 39), existe frequentemente uma fase de pós-processamento em que os resultados da mineração são:

- Inspeccionados – é feita uma leitura dos dados de forma a filtrar informação que possa ser considerada de ruído, ou seja, sem interesse;
- Interpretados – é efectuada uma interpretação dos dados do ponto de vista da informação que contêm, no contexto do domínio da corpora alvo da mineração;
- Avaliados – A qualidade da informação é avaliada utilizando métricas adequadas.

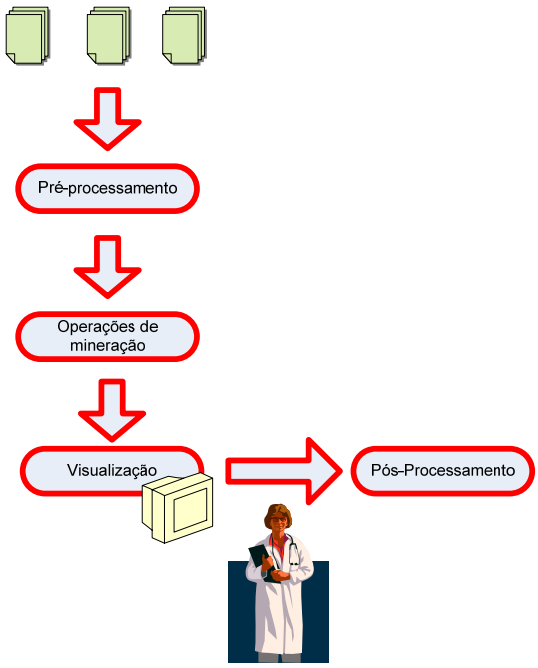

Figura 39 – Actividade final de um pipeline de mineração de texto: pós-processamento.

Nesta fase, consoante os resultados da avaliação, pode eventualmente ser detectada a necessidade de melhorar o sistema de mineração. Neste caso, a inspecção dos dados pode fornecer pistas sobre qual a tarefa ou tarefas de mineração a melhorar.

3. Aplicação prática de conceitos de mineração

Com o objectivo de aplicar e desenvolver algoritmos de mineração de texto, foi pensada uma aproximação prática que permitisse simultaneamente aplicar conceitos genéricos de mineração e tivesse uma potencial utilidade futura, nomeadamente na identificação de documentos "novos" de interesse que fossem sendo retirados diariamente de sistemas online como a PubMed[1], por exemplo. Deste modo, foi decidido desenvolver ferramentas que, com um âmbito principalmente experimental, ajudassem a reunir algumas conclusões sobre a recolha de documentos de interesse de domínios específicos da biomedicina e sobre um modo eventualmente mais eficaz de os representar. Para a questão da representação foi levantada a hipótese de aplicar "comparação" entre corpus, utilizando os termos univocamente pertencentes ao corpus "de interesse" como modelo de representação Adicionalmente, pensou-se em utilizar conhecimento de domínio, de forma que fosse de aplicação simples e controlada (ou seja, programaticamente flexível). Os passos envolvidos num sistema de mineração que tentasse cumprir este objectivo passariam pela aplicação de técnicas de:

- Pré-processamento – atomização, remoção de *stopwords* e *stemming*;
- Representação dos documentos – utilizando um modelo vectorial *bag-of-words* e um método de selecção de atributos;
- Recolha de documentos de interesse – aplicando indexação aos documentos pré-processados e retirando medidas de similaridade euclidiana por pesagem do tipo TF-IDF.
- Anotação – Através do reconhecimento de nomes de entidades do domínio biomédico.

Consequentemente, desenvolveram-se três ferramentas:

- Annotator – Efectua a tarefa de reconhecimento de nomes de entidades utilizando a livraria ABNER, e faz uma posterior filtragem por remoção de todos os outros termos.
- Corparator – Efectua a tarefa de selecção de atributos por comparação entre corpus;
- BDClassifier – Efectua a tarefa de recolha de documentos de interesse com base em cálculos de similaridade euclidiana;

Estas ferramentas são pormenorizadamente descritas nos pontos 3.1.1. a 3.1.3.

Uma vez desenvolvidas estas ferramentas poder-se-iam então levantar algumas hipóteses e levar a cabo experiências que ajudassem a encontrar respostas e compreender melhor as questões envolvidas na aplicação destas técnicas na mineração de literatura biomédica. Estas experiências e a metodologia e resultados encontrados são descritos detalhadamente no ponto 3.2.

3.1. Ferramentas desenvolvidas e utilizadas

Nos pontos a seguir descrevem-se as várias ferramentas desenvolvidas e utilizadas para a aplicação de conceitos de mineração.

3.1.1. O sistema de anotação e filtragem (Annotator)

Com a finalidade de obter uma representação de um corpus baseada no reconhecimento das entidades biomédicas que refere, foi desenvolvido utilizando a linguagem JAVA o sistema "Annotator". Este sistema em termos de reconhecimento de entidades passa por ser uma mera implementação de um cliente da livraria JAVA ABNER [47], um conhecido software *open-source* de reconhecimento de entidades biomédicas. ABNER [47] é uma ferramenta de software para a análise de texto sobre biologia molecular e que utiliza um mecanismo estatístico de aprendizagem máquina e documentos de corpora NLPBA [48] e BioCreative [49] como treino. A escolha desta ferramenta de reconhecimento de entidades biomédicas para utilização no sistema "Annotator" deveu-se a três razões fundamentais:

- No site onde a ferramenta é descrita e disponibilizada (http://pages.cs.wisc.edu/~bsettles/abner/ , em Junho de 2008), é afirmado que possui um desempenho de nível estado da arte;
- É disponibilizada uma JAVA API, *abner.jar*, o que se enquadrava plenamente na linguagem de programação que estávamos a utilizar;
- A implementação de um cliente funcional da API *abner.jar* era simples.

Inicialmente são aplicados ao corpus original os vários passos de anotação que o ABNER permite efectuar, existindo posteriormente uma remoção em todos os documentos de todos os termos que não foram etiquetados como sendo entidades biomédicas. O resultado é então o conjunto inicial de documentos filtrados, com o espaço de atributos reduzido apenas às entidades que o ABNER reconhece (Figura 40).

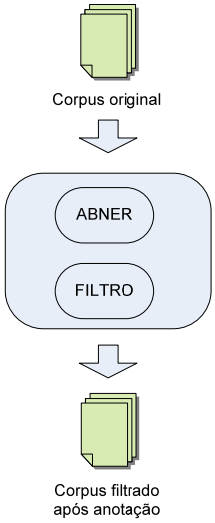

Figura 40 – Esquema genérico de entradas/saídas do sistema Annotator.

Como saídas adicionais, além dos ficheiros de texto referentes aos documentos filtrados, este sistema fornece um conjunto de documentos XML que armazenam de forma estruturada as entidades que foram reconhecidas para cada documento, e que podem eventualmente ser úteis para outras aplicações (para visualização por aplicação de XSLT, por exemplo). Na figura 41 encontra-se um exemplo de ficheiro XML resultante.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<RecognizedEntities>
    <Protein>ribosomal proteins</Protein>
    <Protein>ribosomal proteins</Protein>
    <Protein>typically ribosomal proteins</Protein>
    <Protein>ribosomal proteins</Protein>
    <Dna>nonbiased genomes</Dna>
    <Protein>G+ C</Protein>
    <Dna>G+ C contents of third codon positions</Dna>
    <Dna>G+ C-rich genomes</Dna>
    <Dna>A+ T-rich genomes</Dna>
    <Rna>tRNAs</Rna>
```

Figura 41 – Exemplo de ficheiro XML de saída do sistema "Annotator".

3.1.2. O sistema de selecção de atributos (Corparator – Corpus Comparator)

Na perseguição do objectivo de obtenção de um conjunto de termos representativos de um corpus foi desenvolvido, na linguagem de programação JAVA, um sistema de selecção de atributos que se baseia na filtragem de termos por comparação entre corpus de documentos. Este sistema tem como entradas os conjuntos de documentos correspondentes a cada corpus e tem como saídas conjuntos de termos agrupados em ficheiros, termos esses que são mutuamente exclusivos entre os dois corpus. Ou seja, a saída deste sistema são as listas de termos que pertencem unicamente a cada um dos corpus, sendo eliminados quaisquer termos que eventualmente tivessem em comum (esquema na figura 42 a seguir).

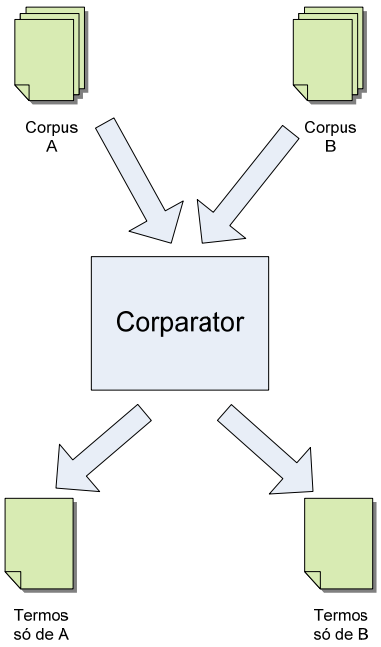

Figura 42 – Esquema genérico de entradas/saídas do sistema Corparator.

Estas listas (de forma a ser possível averiguar qual o critério que eventualmente resulta em melhor desempenho quando as listas são aplicadas como a consulta de entrada num sistema de recolha de documentos de interesse) são geradas consoante limiares de valores de peso dos termos, calculados como:

- Peso TF-IDF (máximo para cada termo);
- Peso CTF;
- Peso CTF * N;

Em que TF (*Term Frequency*) é a frequência do termo, ou número de ocorrências do termo num documento, IDF é a frequência inversa do termo (*Inverse Document Frequency*), CTF (*Corpus Term Frequency*) é o número total de ocorrências do termo em todo o corpus, e N é o número total de documentos presente no corpus em questão em que

o termo ocorre. O peso TF-IDF é o valor correspondente máximo em todo o conjunto de documentos do corpus em que o termo em causa tem ocorrência. Estabelecendo limiares distintos para estes parâmetros é possível obter conjuntos distintos de termos representativos como saída do sistema.

A utilização do sistema pode ser resumida consoante as seguintes etapas:

1. Preenchimento do ficheiro de configuração do sistema (*configuration.properties*) para definição:

- Dos directórios pré-estabelecidos como sendo de entrada do sistema – ou seja, onde são colocados os documentos dos dois corpus (formatos .txt);

- Dos limiares a aplicar;

- Da localização da lista de *stopwords* a utilizar;

- Do tipo de *stemming* desejado (*Porter* [50], *Paice-Husk* [51] ou *Lovins* [52]);

- Dos directórios de saída do sistema – ou seja, directórios onde são escritos os ficheiros com os termos representativos dos corpus (formatos .txt e .csv);

2. Os documentos relativos a cada corpus são depositados nos directórios de entrada;

3. A execução do sistema é efectuada por linha de comando;

4. Procede-se à recolha dos ficheiros resultantes nos directórios de saída (formatos .txt e .csv).

Na figura 43 a seguir pode-se visualizar uma representação deste esquema de utilização.

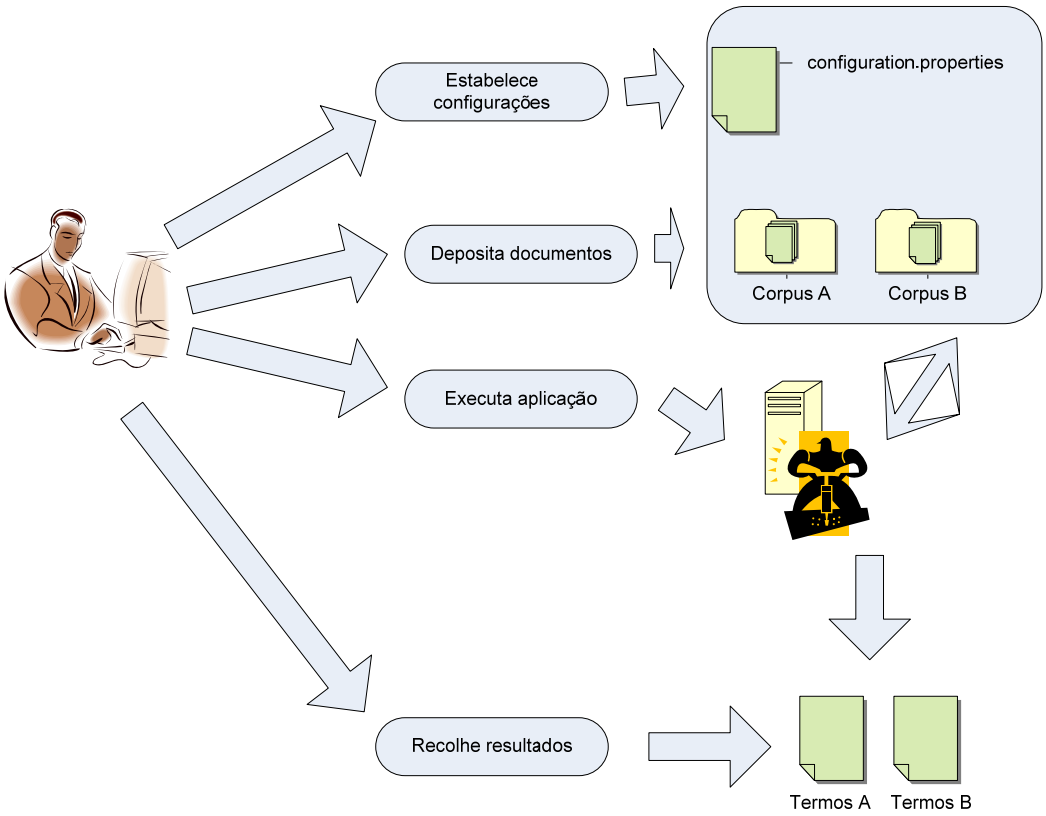

Figura 43 – Esquema de utilização do sistema Corparator.

Funcionamento e arquitectura do sistema

O funcionamento do sistema pode ser resumido da seguinte forma (fig. 44):

1. Análise dos documentos e seu pré-processamento:

    1.1. Atomização;

    1.2. Remoção de palavras sem interesse (*stopwords*);

    1.3. Redução dos termos a uma forma canónica base (*stemming*);

2. Indexação dos termos presentes em cada corpus e construção da representação vectorial dos documentos tendo em conta os critérios de filtragem escolhidos (ou seja, a selecção de atributos a efectuar);

3. Comparação entre os vectores de palavras representativos de cada corpus completo e eliminação dos termos comuns;

4. Escrita dos resultados nos ficheiros correspondentes, nos directórios de saída (formatos .txt e .csv).

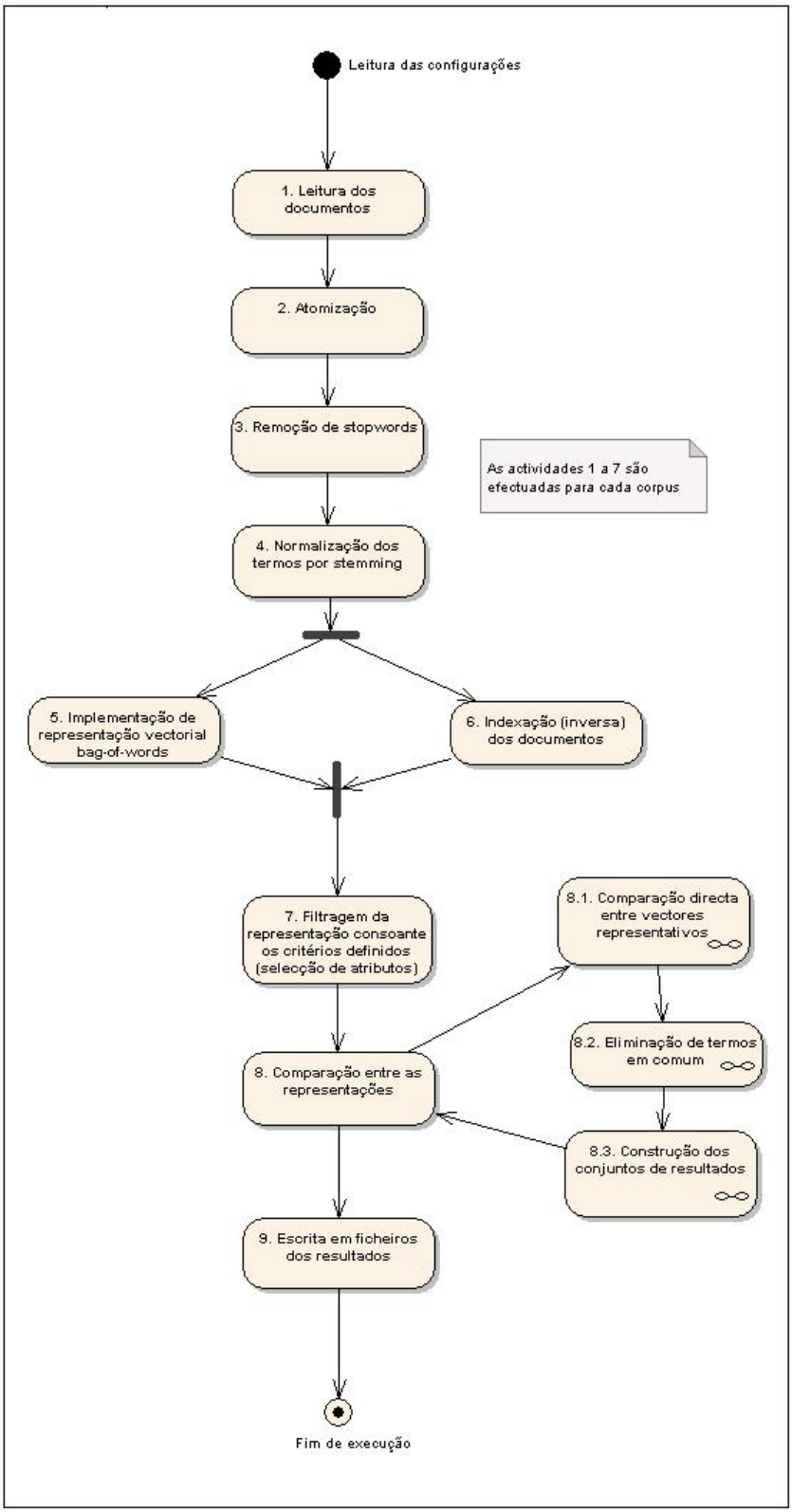

Figura 44 – Diagrama de actividades do sistema Corparator.

Em termos de arquitectura, o sistema é constituído por cinco componentes principais:

- Componente *ieeta.ua.bioinformatics.textmining.application*, que é a componente de interface com o utilizador e a que garante o processamento sequencial das tarefas do sistema. Utiliza na sua execução as quatro outras componentes;

- Componente *ieeta.ua.bioinformatics.textmining.properties,* que é a componente que se responsabiliza pela leitura das configurações estabelecidas para o sistema;

- Componente *ieeta.ua.bioinformatics.textmining.stopwords*, que é a componente responsável pela execução da tarefa de remoção de palavras sem interesse. Esta componente lê o ficheiro que contém a lista de *stopwords* e compara os termos presentes com aqueles que constam nos documentos. Os termos em comum são removidos das representações vectoriais dos documentos;

- Componente *ieeta.ua.bioinformatics.textmining.stemming,* que é a componente que executa a tarefa de redução dos termos a uma forma canónica base. Existem três algoritmos implementados no sistema que podem ser utilizados consoante definição no ficheiro de configuração, nomeadamente os algoritmos de Porter [50], Paice-Husk [51] e Lovins [52];

- Componente *ieeta.ua.bioinformatics.textmining.indexing,* que é a componente que executa a tarefa de indexação dos termos e documentos.

A seguir apresenta-se o respectivo diagrama de colaboração, na figura 45.

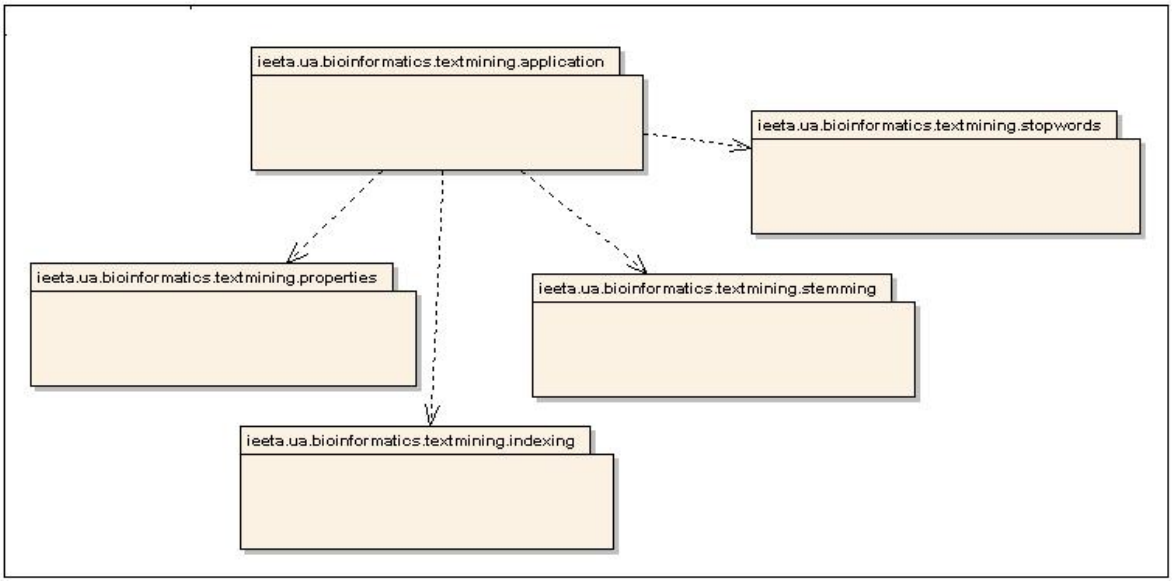

Figura 45 – Diagrama de colaboração do sistema Corparator.

3.1.3. O sistema de recolha de documentos de interesse (BDClassifier – Biomedical Documents Classifier)

De forma a conseguirmos avaliar o desempenho da aplicação das diversas representações obtidas pelo sistema "Corparator" na recolha de documentos de interesse, desenvolvemos o "BDClassifier", também na linguagem JAVA, um classificador de documentos de literatura biomédica.

Este sistema aceita como "treino" a especificação de um conjunto de directórios de entrada, que contêm os ficheiros compostos pelos termos mais representativos dos diversos corpus (cada corpus é visto aqui como representando um assunto ou subdomínio biomédico de interesse). Estes ficheiros de termos, são os ficheiros resultantes do processamento efectuado pelo "Corparator" sobre os corpus originais. Existe uma outra entrada deste sistema que é o directório que contém os documentos do corpus a categorizar. A saída do sistema é um conjunto de ficheiros que listam os documentos classificados e ordenados por valores de similaridade, cada um correspondendo a uma categoria estabelecida. Na figura 46 encontra-se um esquema ilustrativo. Analisando estes ficheiros é possível perceber que documentos foram associados a que categorias e ter uma avaliação de "quão pertencentes" a cada categoria eles são.

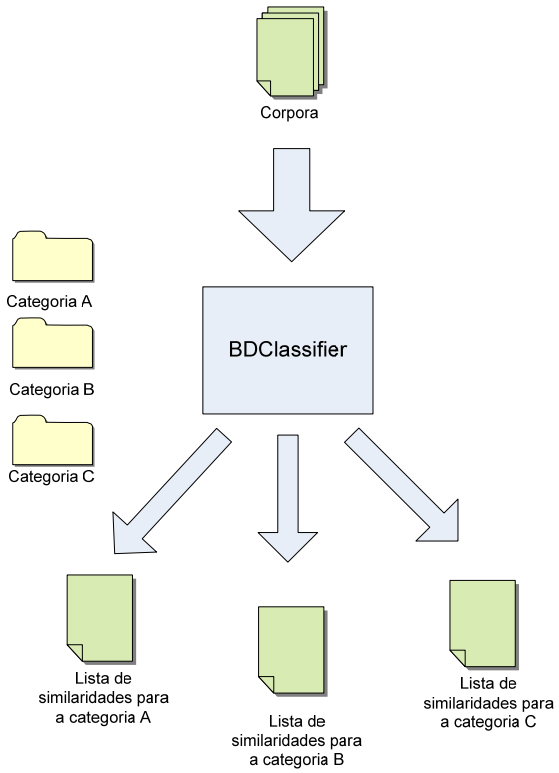

Figura 46 – Esquema genérico de entradas/saídas do sistema BDClassifier.

A fórmula de similaridade utilizada é dada por [27]:

$$Sim(q,d) = \frac{\sum(Wtd * Wtq)}{\|d\| \cdot \|q\|}$$

Onde,

$w_{tq} = f_{tq} * IDF(t)$, é o peso do termo t no "treino" q

$w_{td} = f_{td} * IDF(t)$ é o peso do termo t no documento d.

$\|d\| = \sqrt{\sum (Wtd)^2}$ é o comprimento do documento d.

$\|q\| = \sqrt{\sum (Wtq)^2}$ é o comprimento de q.

A utilização do sistema pode ser resumida consoante as seguintes etapas (fig. 47):

1. Preenchimento do ficheiro de configuração do sistema para definição:

   - Dos directórios pré-estabelecidos como sendo de entrada do sistema – os directórios onde são colocadas as representações dos corpus (as categorias) e o directório onde reside o corpus a categorizar (formatos .txt);

   - Do limiar de similaridade a aplicar;

   - Da localização da lista de *stopwords* a utilizar;

   - Do tipo de *stemming* desejado;

   - Dos directórios de saída do sistema – os directórios onde são escritos os ficheiros com os resultados ordenados de forma decrescente de similaridade (formatos .txt e .csv);

2. As representações de cada categoria são depositadas nos directórios respectivos;

3. O corpus a categorizar é colocado no directório respectivo;

4. A execução do sistema é efectuada por linha de comando;

5. Procede-se à recolha dos ficheiros resultantes nos directórios de saída (formatos .txt e .csv).

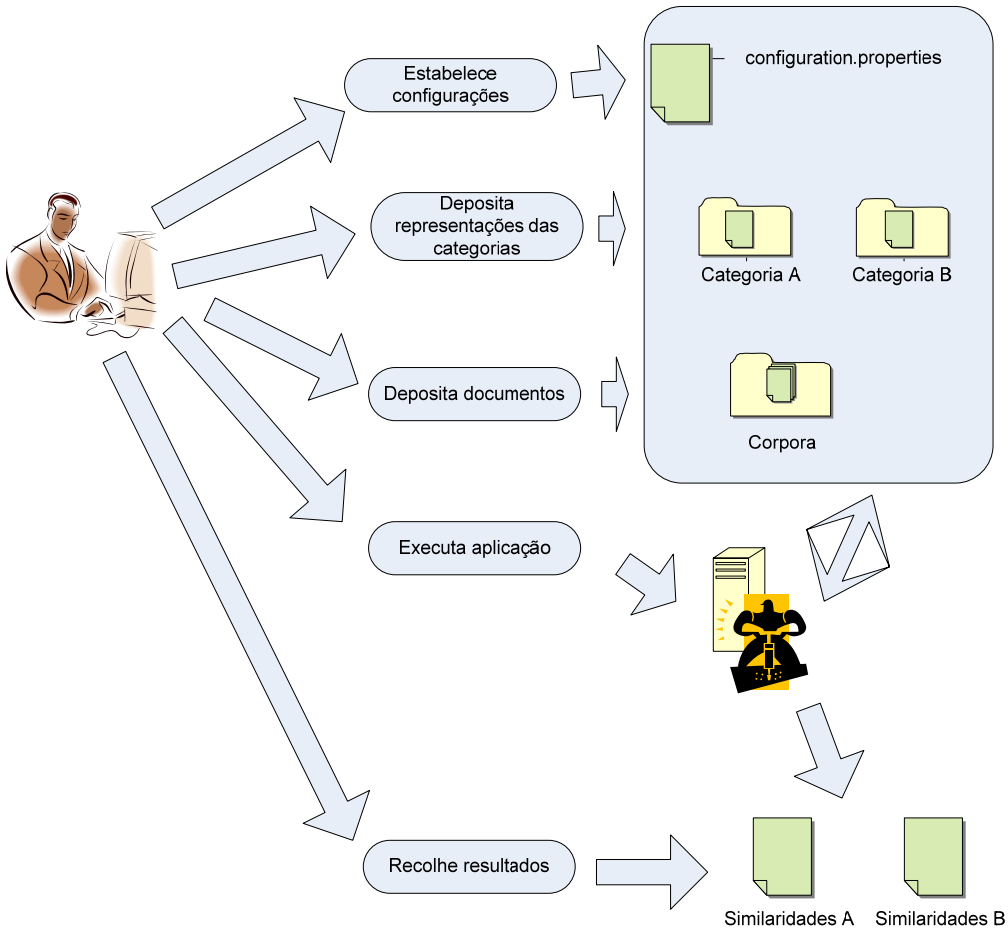

Figura 47 – Esquema de utilização do sistema BDClassifier.

Funcionamento e arquitectura do sistema

O funcionamento do sistema é nos passos de análise dos documentos, pré-processamento e indexação similar ao que acontece no "Corparator". As grandes diferenças residem evidentemente nos passos posteriores. Neste sistema, existe unicamente um método de pesagem, a TF-IDF, e são efectuados cálculos de similaridade entre os documentos a categorizar e aqueles que se encontram nos diversos directórios "treino". Na figura 48 é possível observar o respectivo diagrama de actividades.

Em termos de arquitectura, o sistema é constituído por seis componentes principais, sendo que três das componentes - *ieeta.ua.bioinformatics.textmining.stopwords*, i*eeta.ua.bioinformatics.textmining.stemming*, *ieeta.ua.bioinformatics.textmining.properties* - são comuns às já descritas para o sistema "Corparator". A componente *ieeta.ua.bioinformatics.textmining.indexing* difere por apenas contemplar o peso TF-IDF.

Existe então uma componente adicional *ieeta.ua.bioinformatics.textmining.retrieval* que é a responsável pelos cálculos de similaridade, e uma componente *ieeta.ua.bioinformatics.textmining.application* que embora de igual designação à utilizada para interface no "Corparator", difere obviamente em termos de lógica do processamento.

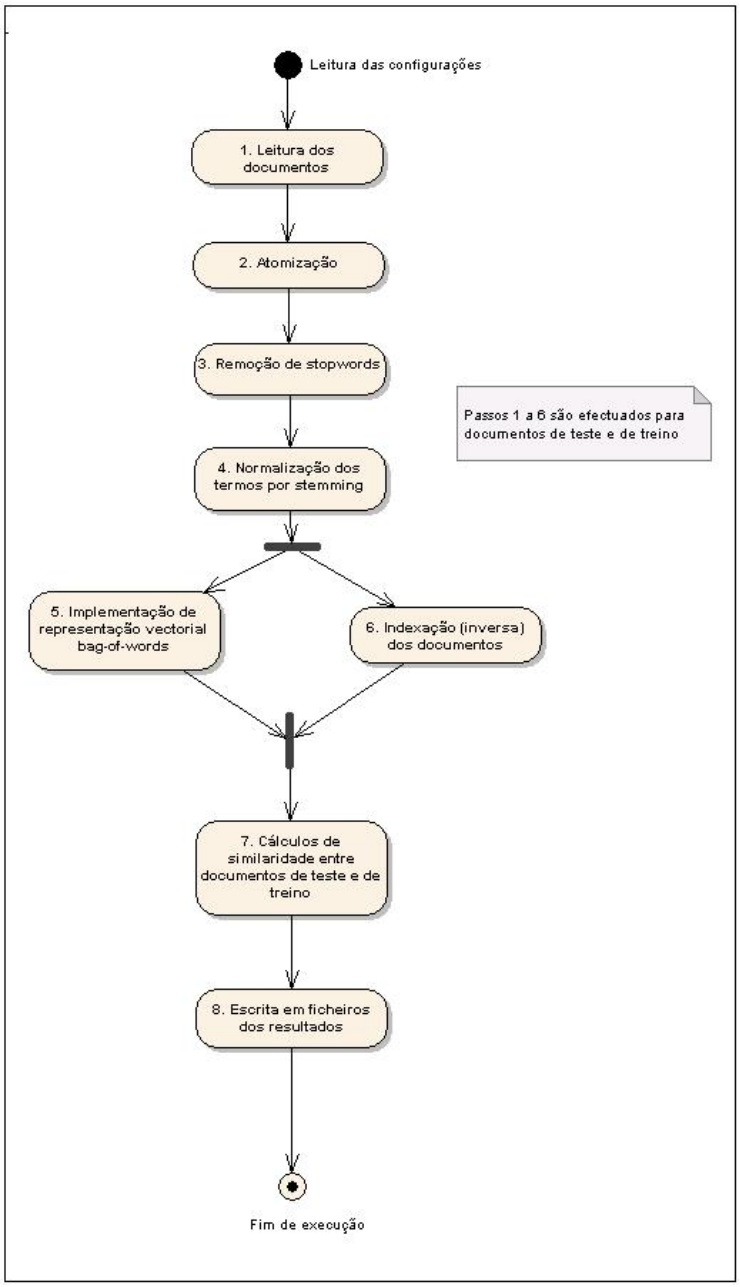

Figura 48 – Diagrama de actividades do sistema BDClassifier.

3.2. Procedimento experimental e resultados obtidos

Utilizando os sistemas desenvolvidos, pretendeu-se encontrar respostas para um conjunto de hipóteses relativas à escolha de um modelo de representação de um corpus de documentos, para aplicação em recolha de documentos de interesse. Nomeadamente:

1. Será que se consegue obter de forma computacionalmente simples um modelo de representação para um corpus que garanta um melhor desempenho quando aplicado num sistema de recolha de documentos?

2. Qual a influência da utilização de pesos de tipos diferentes para os termos na fase de selecção de atributos sobre o desempenho da recolha de documentos, nomeadamente daqueles presentes no sistema Corparator?

    - O peso TF-IDF: Uma forma baseada na importância dos termos que sendo raros no conjunto de documentos se apresentam como pesados ao nível de um documento individual;

    - O peso CTF: Uma forma baseada na importância dos termos que no total ocorrem frequentemente em todo o corpus;

    - O peso CTF * N: Uma forma baseada na importância dos termos que no total ocorrem frequentemente em todo o corpus, num número de documentos, N.

3. Será que há diferenças entre utilizar todo um conjunto de termos representativos de um corpus, ou apenas uma parte (considerada mais significativa) dele, na sua representação?

4. Qual a influência de utilizar na selecção de atributos uma comparação entre um corpus do domínio biomédico e um corpus de domínio genérico ou outro de domínio semelhante?

5. Qual a influência da utilização de reconhecimento de nomes de entidades e respectiva filtragem na fase de selecção de atributos?

Um primeiro passo foi escolher corpora que se considerasse adequada para efectuar experiências, tendo sido escolhidos cinco corpus distintos:

a. Corpus “SEQ_ANALYSIS” – Conjunto de documentos sobre o subdomínio biomédico da análise de sequências (codões) e que foi o utilizado como objecto da análise da selecção de atributos;

b. Corpus “MADATA” – Conjunto de documentos sobre o subdomínio biomédico da análise de dados de *Microarrays*;

c. Corpus "NCB" – Conjunto de documentos (neste caso *abstracts*) heterogéneos do domínio biomédico que constaram da conferência "*XVth National Congress of Biochemistry*" que teve lugar em Aveiro em Dezembro de 2006;

d. Corpus "GENERIC" – Conjunto de documentos de domínio genérico obtido a partir de excertos de obras literárias clássicas em inglês (do projecto *Gutenberg*).

e. Corpus "NEW" – Conjunto de documentos "novos" sobre o subdomínio biomédico da análise de sequências (codões), ou seja, que não têm papel em qualquer operação de selecção de atributos, a utilizar como teste de classificação.

Foi elaborado um conjunto de experiências obedecendo ao seguinte protocolo:

- Utilização dos atributos obtidos para o corpus "SEQ_ANALYSIS" como consulta em BDClassifier, respectivamente:

  - Sem filtragem por critérios de pesagem TF-IDF, CTF e CTF*N;
  - Filtrando pelos 25% de valores mais elevados de TF-IDF, CTF e CTF*N.

Essas experiências realizadas diferiram na forma de filtragem de termos por comparação e foram designadas por:

- SEQ_NO_COMP – em que não se efectuou qualquer filtragem de termos por comparação de "SEQ_ANALYSIS" com outros corpus;

- SEQ_COMP_GEN – em que foi efectuada filtragem através do sistema Corparator por comparação de "SEQ_ANALYSIS" com o corpus de domínio genérico "GENERIC";

- SEQ_COMP_MA_NCB – onde se efectuou filtragem através do sistema Corparator por comparação de "SEQ_ANALYSIS" com os corpus de domínio biomédico "MADATA" e "NCB";

- SEQ_COMP_ALL – em que a filtragem foi feita através do sistema Corparator por comparação de "SEQ_ANALYSIS" com os corpus de domínio biomédico "MADATA" e "NCB" em conjunto com o corpus genérico "GENERIC";

Foi ainda realizada uma experiência em que se utilizou anotação, nomeadamente:

- SEQ_ANNO_FIL – onde a filtragem a "SEQ_ANALYSIS" se efectuou por reconhecimento de nomes de entidades.

Para retirar conclusões quanto à recolha de documentos "novos" foram repetidas as experiências anteriores mas com os documentos do corpus NEW adicionados ao corpus alvo no sistema BDClassifier, numa experiência que se designou por RETRIEVAL.

Por fim, procedeu-se a uma avaliação dos resultados. Como já foi referido no capítulo sobre recolha de documentos de interesse, as medidas mais comuns de avaliação de desempenho de um sistema binário de recolha de documentos são a precisão (*precision*) e a evocação (*recall*), o que fez destas medidas uma escolha lógica como medida da eficácia do desempenho do sistema BDClassifier consoante a selecção de atributos resultante do sistema Corparator. Adicionalmente, devido à natureza "pesada" do tipo de recolha de documentos que queríamos avaliar, foi utilizada a métrica R-*precision* definida como sendo a fracção de documentos SEQ_ANALYSIS presentes no conjunto daqueles que obtiveram os 15 maiores valores de similaridade no sistema BDClassifier, para cada experiência (15, porque é o número de documentos do corpus SEQ_ANALYSIS e uma vez que idealmente a fracção seria igual a 1).

Na figura 49 seguinte é possível ver um esquema que pretende resumir a metodologia seguida para as experiências SEQ_NO_COMP, SEQ_COMP_GEN, SEQ_COMP_MA_NCB e SEQ_COMP_ALL. Na figura 50 está representado esquematicamente o procedimento para a experiência SEQ_ANNO_FIL.

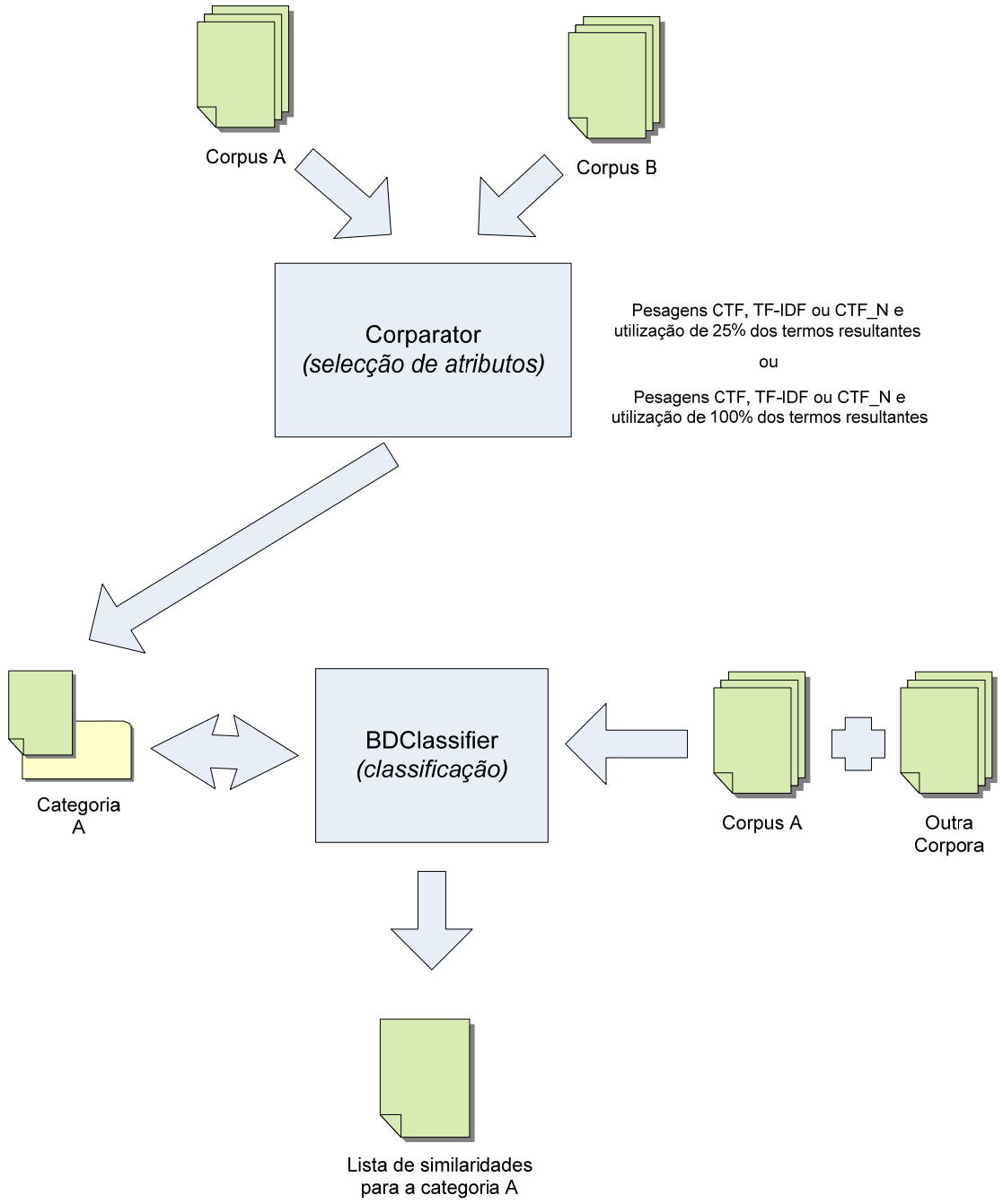

Figura 49 – Diagrama da metodologia experimental sem anotação. É feita uma selecção de atributos para o corpus A por comparação com um outro corpus, utilizando critérios distintos de pesagem dos termos e também de percentagem de utilização dos mesmos. A representação resultante é utilizada para classificar documentos, que incluem os que fazem parte do corpus A. Os resultados são então recolhidos e avaliados.

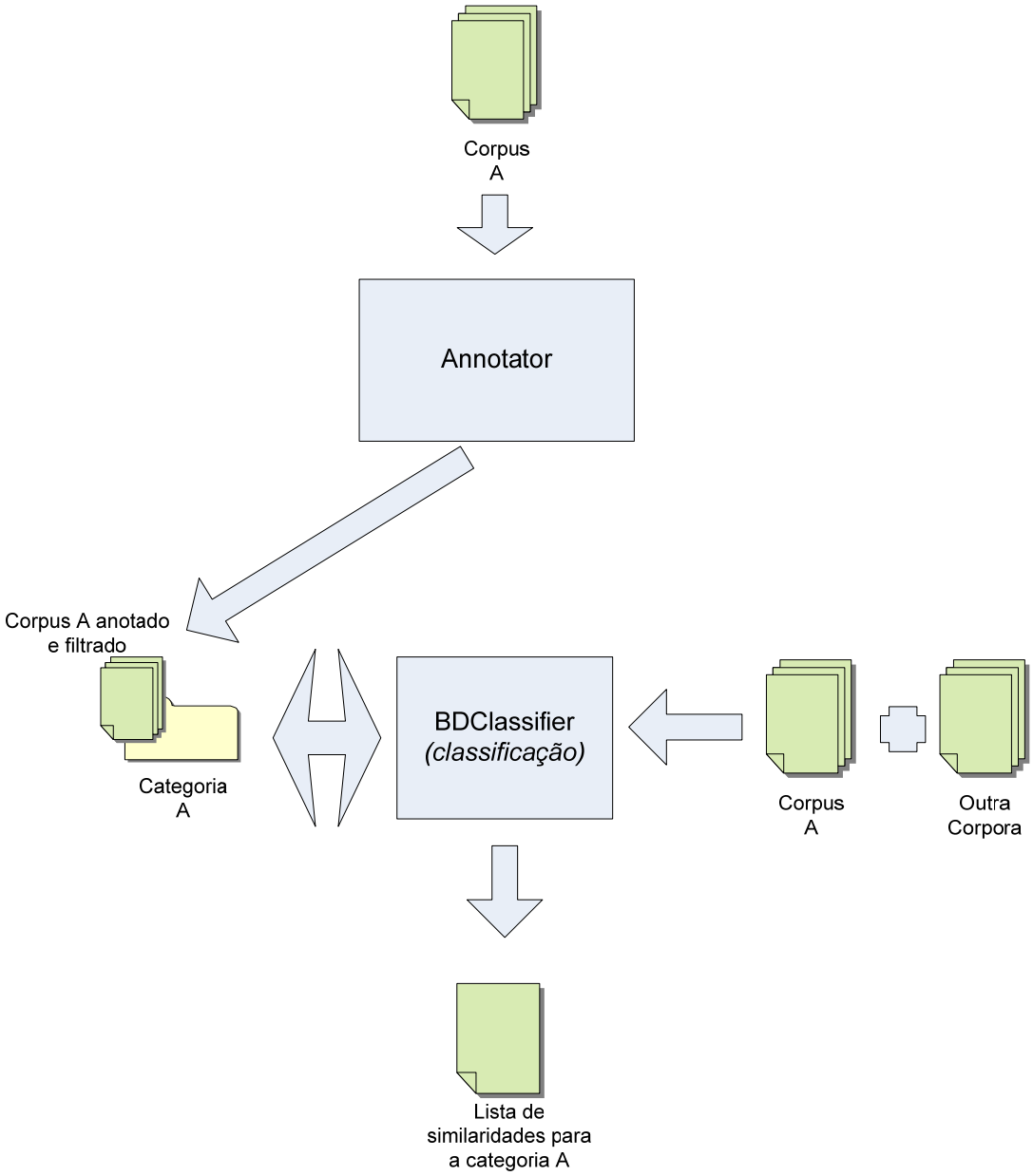

Figura 50 – Diagrama da metodologia experimental com anotação. O corpus é anotado e os termos "a mais", ou seja aqueles que não foram reconhecidos como entidades biomédicas de interesse são removidos dos documentos. O corpus assim modificado é então utilizado para classificar documentos, que incluem os que fazem parte do corpus A. Os resultados são então recolhidos e avaliados.

3.2.1. Apresentação e discussão dos resultados obtidos.

A seguir são apresentados e discutidos os resultados alcançados nas diversas experiências, utilizando as designações acima definidas.

SEQ_NO_COMP

Experiência	% Atributos	Tipo de Pesagem	Precision	Recall	*R-precision*	Similaridade Média (SEQ_ANALYSIS)
SEQ_NO_COMP	25	TF-IDF	0.114	1	0.8	0.191148
		CTF_N	0.114	1	0.53(3)	0.238511
		CTF	0.114	1	0.6	0.233097
	100	TF-IDF	0.114	1	0.6	0.149588
		CTF_N	0.114	1	0.6	0.138239
		CTF	0.114	1	0.6	0.138239

Tabela 13 – Resumo dos resultados obtidos de Precision e Recall.

Verifica-se analisando a tabela 13 que utilizando todos os termos presentes nos documentos do corpus SEQ_ANALYSIS, sem qualquer redução por comparação com outro corpus, ou por selecção dos termos com maior peso, os valores para a medida *R-precision* obtidos são baixos. Contudo, na situação em que se seleccionam os 25% dos termos presentes no corpus SEQ_ANALYSIS com pesos TF-IDF mais significativos, esses valores de sobem, indicando uma clara melhoria nos resultados da recolha de documentos de interesse.

Isto deve-se a:

- O facto dos termos com peso TF-IDF mais elevado serem aqueles que mais identificam a diferença do teor do domínio dos documentos do corpus SEQ_ANALYSIS relativamente aos documentos dos outros corpus. Como tal, quando se efectuam os cálculos de similaridade no sistema BDClassifier representando o corpus pelos seus termos de TF-IDF mais significativos, e apenas por estes, a probabilidade dos termos diferentes de zero presentes na fórmula de cálculo pertencerem a documentos de natureza comum ao corpus é superior, e os resultados da medida *R-precision* aumentam (melhoram);

- Quando são utilizados todos os termos presentes no corpus, estão a ser utilizados nos cálculos de similaridade termos da linguagem comum que surgem como significativos em termos de pesagem. Assim, a similaridade com documentos externos ao domínio do corpus SEQ_ANALYSIS aumenta resultando numa deterioração dos valores da medida *R-precision*.

Os resultados detalhados desta experiência encontram-se no Anexo A.

SEQ_COMP_GEN

Experiência	% Atributos	Tipo de Pesagem	Precision	Recall	*R-prec.*	Similaridade Média (SEQ_ANALYSIS)
SEQ_COMP_GEN	25	TF-IDF	0.114	1	0.86(6)	0.178825
		CTF_N	0.122	1	0.86(6)	0.188128
		CTF	0.122	1	0.8	0.190999
	100	TF-IDF	0.114	1	0.86(6)	0.178825
		CTF_N	0.122	1	0.73(3)	0.124120
		CTF	0.122	1	0.73(3)	0.124120

Tabela 24 – Resumo dos resultados obtidos de Precision e Recall.

Verifica-se analisando a tabela 24 que utilizando os termos resultantes da comparação no sistema Corparator do corpus SEQ_ANALYSIS com o corpus GENERIC como consulta no sistema BDClassifier, os valores de *R-precision* obtidos são genericamente mais elevados que os obtidos na experiência SEQ_NO_COMP. Mais, verifica-se que os melhores resultados (*R-precision*) se verificam quando se seleccionam os 25% dos termos presentes no corpus SEQ_ANALYSIS com pesos TF-IDF, CTF_N mais significativos, ou quando se utilizam todos os termos, mas com TF-IDF não nulo.

Isto deve-se a:

- Quando se efectua a comparação com o corpus GENERIC, todos os termos que estejam presentes em ambos os corpus são removidos da representação do corpus SEQ_ANALYSIS. Ou seja, como os termos do corpus GENERIC são da linguagem inglesa comum, no fundo está-se a forçar um efeito de remoção de *stopwords*, como se estivéssemos a utilizar uma lista de *stopwords* muito mais completa. Deste modo, os termos que sobram, e devido aos outros corpus utilizados como alvo da classificação serem do domínio biomédico, são expectavelmente na sua maioria específicos da biomedicina. Consequentemente os cálculos de similaridade com documentos do domínio biomédico são aqueles que têm resultados significativos e os valores de *R-precision* para a recolha de documentos SEQ_ANALYSIS melhoram;

- A pesagem TF-IDF é aquela que obtém melhores resultados pois tal como referido anteriormente na experiência SEQ_NO_COMP, é aquela que resulta numa selecção de termos de maior significância, agora com a vantagem destes termos terem uma menor probabilidade de serem do inglês comum.

Os resultados detalhados desta experiência podem ser consultados no Anexo B.

SEQ_COMP_MA_NCB

Experiência	% Atributos	Tipo de Pesagem	Precision	Recall	*R-pre.*	Similaridade Média (SEQ_ANALYSIS)
SEQ_COMP_MA_NCB	25	TF-IDF	0.636	0.93(3)	0.93(3)	0.179247
		CTF_N	0.636	0.93(3)	0.93(3)	0.198629
		CTF	0.636	0.93(3)	0.93(3)	0.197207
	100	TF-IDF	0.636	0.93(3)	0.86(6)	0.149856
		CTF_N	0.636	0.93(3)	0.86(6)	0.149856
		CTF	0.636	0.93(3)	0.86(6)	0.149856

Tabela 35 – Resumo dos resultados obtidos de Precision, Recall e *R-precision*.

Verifica-se analisando a tabela 35 que utilizando os termos resultantes da comparação no sistema Corparator do corpus SEQ_ANALYSIS com a corpora resultante da adição dos corpus MADATA e NCB como consulta no sistema BDClassifier, os valores de *R-precision* obtidos são genericamente mais elevados que nas experiências anteriores. Mais, verifica-se também que quando se seleccionam os 25% dos termos presentes no corpus SEQ_ANALYSIS com pesos TF-IDF, CTF_N ou CTF mais significativos, os valores de *R-precision* sobem, embora não de forma muito significativa. Curiosa é a descida do valor de *recall* relativamente às experiências anteriores e a subida do valor de *precision*.

Isto deve-se a:

- Quando se efectua a comparação com os corpus MADATA e NCB, todos os termos que estejam presentes em ambos os corpus são removidos da representação do corpus SEQ_ANALYSIS. Todos estes três corpus possuem termos da linguagem inglesa comum, e tal como acontecia no caso anterior com a comparação com o corpus GENERIC, está-se a forçar um efeito de remoção de *stopwords*, como se estivéssemos a utilizar uma lista de *stopwords* mais completa. Mas, agora, adicionalmente, estão a ser removidos os termos comuns também do domínio biomédico. Deste modo, seria expectável que os termos que sobram na sua maioria fossem específicos da biomedicina, nomeadamente do subdomínio focado pelo corpus SEQ_ANALYSIS. No entanto verifica-se que estão a ser recolhidos no sistema BDClassifier documentos do corpus GENERIC. Isto dever-se-á a "sobrarem" da comparação ainda como representativos do corpus SEQ_ANALYSIS muitos termos do inglês comum. Consequentemente, apesar dos valores de *R-precision* para a recolha de documentos SEQ_ANALYSIS serem bons verifica-se pontualmente uma similaridade com documentos que nem sequer fazem parte do domínio biomédico;

- A descida do valor de *recall* deve-se à ausência nos documentos recolhidos de um documento do corpus SEQ_ANALYSIS ("CodonUsageAll.txt"). Esse documento é constituído praticamente por umas poucas frases e resultados numéricos. O que provavelmente aconteceu foi que nos outros corpus de domínio biomédico estava presente também o termo "chave" (no caso, *codon*) constituinte das poucas frases que o documento em causa continha, tendo sido portanto removido da representação do corpus. Logo, aquando dos cálculos de similaridade e devido à ausência deste termo, o documento não tem associado qualquer valor de similaridade, ficando ausente na recolha.

- A subida do valor de *precision* prende-se com o facto dos documentos recolhidos pertencerem apenas aos corpus SEQ_ANALYSIS e GENERIC, que são ambos pequenos, descendo assim bastante o valor do numerador da fracção de cálculo.

No Anexo C encontram-se os vários resultados detalhados.

SEQ_COMP_ALL

Experiência	% Atributos	Tipo de Pesagem	Precision	Recall	*R-precision*	Similaridade Média (SEQ_ANALYSIS)
SEQ_COMP_ALL	25	TF-IDF	1	0.93(3)	0.93(3)	0.178331
		CTF_N	1	0.93(3)	0.93(3)	0.197267
		CTF	1	0.93(3)	0.93(3)	0.195322
	100	TF-IDF	1	0.93(3)	0.93(3)	0.144070
		CTF_N	1	0.93(3)	0.93(3)	0.144070
		CTF	1	0.93(3)	0.93(3)	0.144070

Tabela 44 – Resumo dos resultados obtidos de Precision, Recall e *R-precision*.

Verifica-se analisando a tabela 44 que utilizando os termos resultantes da comparação no sistema Corparator do corpus SEQ_ANALYSIS com a corpora resultante da adição dos corpus MADATA, NCB e GENERIC como consulta no sistema BDClassifier, os valores de *recall* e *R-precision* obtidos são elevados e que se obtém um valor óptimo de *precision* igual a um. Mais, verifica-se também que quando se seleccionam os 25% dos termos presentes no corpus SEQ_ANALYSIS com pesos TF-IDF, CTF_N ou CTF mais significativos, os valores de *precision*, *recall* e *R-precision* mantém-se, verificando-se um ligeiro aumento nas similaridades médias.

Isto deve-se a:

- Quando se efectua a comparação com os corpus MADATA, NCB e GENERIC, todos os termos que estejam presentes nestes três corpus que

sejam comuns aos do corpus SEQ_ANALYSIS são removidos da sua representação. Deste modo, os termos que sobram são na sua totalidade do sub-domínio focado pelo corpus SEQ_ANALYSIS e serão os únicos significativos nos cálculos de similaridade.

- A redução do conjunto de atributos representativos por filtragem dos 25% mais pesados não implica qualquer variação no desempenho pois só termos de SEQ_ANALYSIS estão envolvidos no cálculo de similaridade, logo, não interessa se são todos ou só parte deles.

- O valor óptimo de 1 para *precision*, prende-se com o facto de no sistema BDClassifier serem recolhidos 14 documentos com similaridade superior a zero, todos eles do corpus SEQ_ANALYSIS, ou seja, sem a presença de qualquer documento externo a esse corpus.

- O valor de *recall* tem a mesma justificação que a dada na experiência anterior.

No Anexo D é possível consultar os resultados detalhados.

SEQ_ANNO_FIL

Precision	Recall	*R-prec.*	Similaridade Média (SEQ_ANALYSIS)
0.114	1	0.93(3)	0.419250

Tabela 47 – Resultados obtidos para SEQ_ANNO_FIL.

Analisando os resultados verifica-se que a utilização do corpus anotado e filtrado pelo sistema Annotator, como consulta no sistema BDClassifier leva a valores elevados de *R-precision* e *recall*, mas baixos de *precision*. Os valor elevado de *R-precision* deve-se ao facto do corpus SEQ_ANALYSIS ser sobre um subdomínio biomédico específico e portanto focar entidades que muito provavelmente não estão presentes nos documentos dos outros corpus. Contudo, o valor baixo de *precision* indica que são também recolhidos muitos documentos dos outros corpora. Uma análise ao que poderia eventualmente provocar este comportamento resultou na identificação de vários reconhecimentos mal efectuados por parte do ABNER, que resultaram em atributos seleccionados não específicos da área da biomedicina, o que teve consequentemente influencia nos cálculos de similaridade.

No Anexo E estão detalhados os resultados alcançados.

Comparação dos resultados sem documentos "novos"

A comparação dos resultados acima apresentados para as diversas experiências, permite retirar algumas conclusões.

Experiência	% Atributos	Tipo de Pesagem	Precision	Recall	*R-prec.*	Similaridade Média (SEQ_ANALYSIS)
SEQ_NO_COMP	25	TF-IDF	0.114	1	0.8	0.191148
		CTF_N	0.114	1	0.53(3)	0.238511
		CTF	0.114	1	0.6	0.233097
	100	TF-IDF	0.114	1	0.6	0.149588
		CTF_N	0.114	1	0.6	0.138239
		CTF	0.114	1	0.6	0.138239
SEQ_COMP_GEN	25	TF-IDF	0.114	1	0.86(6)	0.178825
		CTF_N	0.122	1	0.86(6)	0.188128
		CTF	0.122	1	0.8	0.190999
	100	TF-IDF	0.114	1	0.86(6)	0.178825
		CTF_N	0.122	1	0.73(3)	0.124120
		CTF	0.122	1	0.73(3)	0.124120
SEQ_COMP_MA_NCB	25	TF-IDF	0.636	0.93(3)	0.93(3)	0.179247
		CTF_N	0.636	0.93(3)	0.93(3)	0.198629
		CTF	0.636	0.93(3)	0.93(3)	0.197207
	100	TF-IDF	0.636	0.93(3)	0.86(6)	0.149856
		CTF_N	0.636	0.93(3)	0.86(6)	0.149856
		CTF	0.636	0.93(3)	0.86(6)	0.149856
SEQ_COMP_ALL	25	TF-IDF	1	0.93(3)	0.93(3)	0.178331
		CTF_N	1	0.93(3)	0.93(3)	0.197267
		CTF	1	0.93(3)	0.93(3)	0.195322
	100	TF-IDF	1	0.93(3)	0.93(3)	0.144070
		CTF_N	1	0.93(3)	0.93(3)	0.144070
		CTF	1	0.93(3)	0.93(3)	0.144070
SEQ_ANNO_FIL	-	-	0.114	1	0.93(3)	0.419250

Tabela 48 – Súmula dos resultados obtidos nas várias experiências sem introdução no corpus alvo de classificação por parte do sistema BDClassifier dos documentos do corpus "NEW".

Analisando a tabela 48 conclui-se que:

- A selecção de atributos utilizando comparação por parte do sistema Corparator resulta genericamente em melhorias em termos da recolha de documentos – valores de *R-precision*, *precision* e *recall*. Note-se na diferença entre os resultados obtidos para a experiência SEQ_NO_COMP e as restantes que utilizam comparação;

- É importante a escolha dos corpus para comparação consoante o domínio do conjunto de documentos a classificar. Notem-se as diferenças entre os resultados das experiências SEQ_COMP_GEN, SEQ_COMP_MA_NCB e SEQ_COMP_ALL.

- A redução do número de atributos representativos de um corpus deve ser feita tendo em atenção o método de selecção de atributos que foi realizado. Notem-se as diferenças (embora ligeiras) entre os valores obtidos nas diversas experiências em que existiu comparação;

- A anotação seguida de filtragem aparenta ser uma forma simples de representar um corpus de um dado domínio biomédico, com bons resultados em termos de *R-precision*, tendo em conta os resultados obtidos na experiência SEQ_ANNO_FIL.

RETRIEVAL

Para aferir da capacidade de recolha de documentos "novos", ou seja, que não fazem parte do corpus SEQ_ANALYSIS, mas cujo domínio biomédico é semelhante, repetiram-se as experiências anteriores, mas introduzindo o corpus NEW na corpora alvo a classificar no sistema BDClassifier. Achamos que seria repetitivo e sem utilidade relevante fazer uma descrição detalhada dos resultados das experiências, optando por apresentar a tabela 49, que contém uma súmula.

Comparação dos resultados com documentos "novos"

Experiência	% Atributos	Tipo de Pesagem	Similaridade Média (NEW)	Documentos novos recolhidos
SEQ_NO_COMP	25	TF-IDF	0.0	0
		CTF_N	0.0	0
		CTF	0.0	0
	100	TF-IDF	0.0	0
		CTF_N	0.0	0
		CTF	0.0	0
SEQ_COMP_GEN	25	TF-IDF	0.0	0
		CTF_N	0.0	0
		CTF	0.0	0
	100	TF-IDF	0.0	0
		CTF_N	0.0	0
		CTF	0.0	0
SEQ_COMP_MA_NCB	25	TF-IDF	0.070086	2
		CTF_N	0.099864	2
		CTF	0.107585	1
	100	TF-IDF	0.067397	1
		CTF_N	0.067397	1
		CTF	0.067397	1
SEQ_COMP_ALL	25	TF-IDF	0.060316	3
		CTF_N	0.097781	3
		CTF	0.086064	3
	100	TF-IDF	0.057828	3
		CTF_N	0.057828	3
		CTF	0.057828	3
SEQ_ANNO_FIL	-	-	0.317417	3

Tabela 49 – Súmula dos resultados obtidos nas várias experiências, com os documentos novos adicionados ao corpus alvo do sistema BDClassifier.

Na tabela 49, os "Documentos novos recolhidos", são aqueles que pertencendo ao corpus NEW, obtiveram resultados de similaridade entre os 18 maiores no sistema BDClassifier, utilizando como consulta a representação do corpus SEQ_ANALYSIS em cada situação.

Fazendo uma análise dos resultados obtidos conclui-se que:

- Estão coerentes com as conclusões que tinham sido retiradas nas experiências anteriores, ou seja, a qualidade dos resultados obtidos de *R-precision* reflectiu-se na capacidade de recolha dos documentos "novos";

- A selecção de atributos por comparação, para representação de um corpus e aplicação num sistema de recolha de documentos de interesse, garante bons resultados na recolha de documentos "novos", utilizando um máximo de corpora extra subdomínio. Ou seja, é desejável que tendo uma dada corpora a classificar, e querendo representar um subdomínio, seja utilizado na filtragem e selecção de termos por comparação o maior número de documentos possível;

- A selecção de atributos por anotação e filtragem além de garantir uma recolha correcta de documentos "novos", garante maiores valores de similaridade.

4. Conclusão

Ao longo desta dissertação foram apresentadas as diversas técnicas e metodologias envolvidas na mineração de texto, contextualizando-as numa organização genérica de um processo de mineração. Foram detalhadas as fases envolvidas, nomeadamente pré-processamento, operações nucleares de mineração, visualização e pós-processamento. Ao pré-processamento e respectivas actividades constituintes foi dado um especial ênfase, devido à sua reconhecida importância e a ser a fase sobre a qual incidia a componente prática de estudo e desenvolvimento.

Com vista a obter um sistema eficaz de recolha de documentos de subdomínios biomédicos, minimizando a utilização de ferramentas já existentes no mercado e maximizando a aplicação de conceitos e consequente aprendizagem, foram desenvolvidas três ferramentas de software, de índole maioritariamente experimental.

Estas ferramentas, desenvolvidas na linguagem de programação JAVA, permitiram não só o desenvolvimento e aplicação de algoritmos de atomização, normalização por *stemming*, remoção de *stopwords*, indexação de documentos, pesagem de termos, selecção de atributos, representação vectorial, cálculo de similaridade entre documentos e anotação, mas também efectuar algumas experiências que resultaram nalgumas conclusões sobre formas de filtragem de termos para representação de um corpus e posterior aplicação em recolha de documentos de interesse.

A primeira ferramenta, denominada de "Annotator", utilizando a livraria ABNER [48] é capaz de executar a tarefa de anotação por reconhecimento de entidades de interesse de um corpus e aplicar uma posterior filtragem aos documentos. O resultado é uma representação de documentos em que os termos constituintes não são mais do que os nomes que o ABNER foi capaz de reconhecer, tais como genes e proteínas, por exemplo.

O "Corparator", de maior complexidade, efectua sobre dois corpus distintos os diversos passos de pré-processamento acima descritos e compara as suas representações – entendendo-se por "representação" o conjunto de termos normalizados que constituem o conjunto de documentos do corpus. Como resultado obtêm-se os termos que pertencem univocamente a cada corpus, em ficheiros separados, e consoante três esquemas de pesagem: TF-IDF (*term frequency - inverse document frequency*), CTF (*corpus term-frequency*) e TF*N (em que simplesmente se calcula o peso de um termo não só pela sua *corpus term-frequency*, mas também pelo número de documentos em que aparece, ou seja, pela *document-frequency*).

Por último, a ferramenta "BDClassifier" permite a execução simultânea de consultas (que podem também ser vistas como "treino"), definidas por sequências de termos, e retirar resultados de similaridade com corpus de documentos a classificar (ou recolher). Esta ferramenta, tal como o "Corparator" efectua sobre os documentos e consultas os vários passos de pré-processamento, utilizando posteriormente as respectivas representações vectoriais para obter cálculos euclidianos de similaridade.

Estas ferramentas desenvolvidas permitiram obter resultados em termos de métricas *recall*, *precision* e R-*precision* e retirar conclusões sobre o método de selecção de atributos escolhido e efeitos da sua aplicação em consultas para a recolha de documentos de interesse "novos", no sistema "BDClassifier". Ou seja, documentos que não fazendo parte do corpus originalmente utilizado (representado) como consulta do classificador, se situam em termos de conteúdo semântico num subdomínio biomédico próximo.

Como trabalho futuro, e com base nos resultados obtidos, fica a sugestão de desenvolvimento de uma solução integrada com os sistemas "Corparator" e "BDClassifier", com vista obter o melhor dos dois mundos: ter um sistema que é capaz de gerar representações de corpus de forma dinâmica, alimentando-se dos próprios documentos que vão sendo recolhidos, com base em limiares de similaridade. Tal sistema poderá assim ir "aprendendo" e melhorando cada vez mais as representações que possui para os diversos corpus de contextos biomédicos distintos de interesse, resultando eventualmente em melhor desempenho.

Alternativamente, poder-se-á aprofundar e melhorar o desenvolvimento do sistema "Annotator", utilizando conhecimento específico do domínio biomédico e aplicando anotação e reconhecimento de entidades biomédicas, de forma a obter representações dos documentos de tamanho mais reduzido e conteúdo semântico importante. Contudo para esta solução ter-se-á de confiar em desenvolvimento aplicacional alheio.

Referências

[1] “PubMed”, http://www.ncbi.nlm.nih.gov/pubmed/, online on April 2008.

[2] M. Rajman and R. Besançon, "Text Mining Natural Language Techniques and Text Mining Applications," presented at Seventh IFIP 2.6 Working Conference on Database Semantics (DS-7), Leysin, Switzerland, 1997.

[3] T. Ah-H, "Text Mining: The state of the art and the challenges," presented at Pacific Asia Conf on Knowledge Discovery and Data Mining (PAKDD'99), 1999.

[4] A. Hotho, A. N¨urnberger, and G. Paaß, "A Brief Survey of Text Mining," 2005.

[5] A. Hearst Marti, "Untangling Text Data Mining," presented at ACL '99: 37º Annual Meeting of the Association for Computational Linguistics, University of Maryland, 1999.

[6] R. Feldman and I. Dagan, "Knowledge Discovery in Textual Databases," presented at First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, 1995.

[7] Y. Kodratoff, "About Knowledge Discovery in Texts: A Definition and an Example," Univ. Paris-Sud,, 2000.

[8] K. e. H. Schnattinger, Udo "Intelligent Text Analysis For Dynamically Maintaining And Updating Domain Knowledge Bases " Lecture Notes in Computer Science, 1997.

[9] M. Hearst, "What is Text Mining?" SIMS UC Berkeley 2003.

[10] R. Feldman, J.Sanger, "The text mining handbook, advanced approaches in analyzing unstructured data", Cambridge University Press 2007

[11] Kjell-Inge Skogstad et Al.. "Mining association rules in temporal document collections", Proceedings the 16th International Symposium on Methodologies for Intelligent Systems (ISMIS'2006), Bari, Italy, September 2006.

[12] Mitchell P. Marcus et Al., "Building a Large Annotated Corpus of English: The Penn Treebank", Computational Linguistics volume 19, number 2, pages 313-330, 1994

[13] "The University of Pennsylvania (Penn) Treebank Tag-set", http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html, online on July 2008

[14] Santorini, B. 1990. "Part-of-speech tagging guidelines for the Penn Treebank Project." Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.

[15] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging," Computational Linguistics, vol. 21, pp. 543-565, 1995.

[16] B. Thorsten, "TnT: a statistical part-of-speech tagger," in Proceedings of the sixth conference on Applied natural language processing. Seattle, Washington: Morgan Kaufmann Publishers Inc., 2000.

[17] A. Ratnaparkhi, E. Brill, and K. Church, "A Maximum Entropy Model for Part-of-Speech Tagging," in Proceedings of the Conference on Empirical Methods in Natural Language Processing: Association, 1996, pp. 133-142.

[18] J. Giménez and L. Marquez, "Fast and accurate part-of-speech tagging: The SVM approach revisited," presented at Recent Advances in Natural Language Processing Borovets, Bulgaria, 2003.
[19] Brill, Eric "Some Advances in Transformation-Based Part of Speech Tagging", National Conference on Artificial Intelligence, pages 722-727, 1994
[20] H. Udo and W. Joachim, "High-desempenho tagging on medical texts," in Proceedings of the 20th international conference on Computational Linguistics. Geneva, Switzerland: Association for Computational Linguistics, 2004.
[21] do Prado, H.A. & Ferneda, Edilson, "Emerging technologies of text mining", Information Science Reference, New York 2008
[22] Salton G., & McGill, M.J., "Introduction to modern information retrieval", McGraw Hill Book Company, New York 1983
[23] W. B. Michael, Survey of Text Mining: Springer-Verlag New York, Inc., 2003.
[24] L. Auvil and D. Searsmith, "Using Text Mining for Spam Filtering," presented at Super Computing 2003 (SC2003), Phoenix, 2003.
[25] K.S. Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of Documentation, vol. 28, pp. 11-21, 1972.
[26] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," Journal of Documentation, vol. 60, pp. 503-520, 2004.
[27] Manning Christopher & Prabhakar Raghavan & Hinrich Schutze, "Introduction to information retrieval", Cambridge University Press, New York 2008
[28] Yang, Y., and Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. In Proceedings of ICML-97, 14th International Conference on Machine Learning. D. H. Fisher. Nashville, TN, Morgan Kaufmann Publishers, San Francisco: 412–420.
[29] Baker, L. D., and McCallum, A. K. (1998). Distributional Clustering of Words for Text Classification. In Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval. Melbourne, Australia, ACM Press, New York: 96–103.
[30] Slonim, N., and Tishby, N. (2001). The Power of Word Clusters for Text Classification. In Proceedings of ECIR-01, 23rd European Colloquium on Information Retrieval Research. Darmstadt, Germany Academic Press, British Computer Society, London.
[31] Lewis,D.D. (1992a).AnEvaluation of Phrasal and Clustered Representations on aText Categorization task. In Proceedings of SIGIR-92, 15thACMInternational Conference on Research and Development in Information Retrieval. N. Belkin, P. Ingwersen, and A. M. Pejtersen, eds. Copenhagen, ACM Press, New York: 37–50.
[32] Lewis, D. D. (1992b). Representation and Learning in Information Retrieval. Ph.D.thesis, Department of Computer Science, University of Massachusetts.
[33] Li, Y. H., and Jain, A. K. (1998). "Classification of Text Documents." The Computer Journal 41(8): 537–546.
[34] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," Journal of the American Society for Information Science, vol. 41, pp. 391-407, 1990.
[35] M. Hearst. "Untangling text data mining." In Proc. of ACL'99 the 37th Annual Meeting of the Association for Computational Linguistics, 1999.

[36] D. D. Lewis, "Evaluating text categorization," in Proceedings of the workshop on Speech and Natural Language. Pacific Grove, California: Association for Computational Linguistics, 1991.
[37] N. Kamal, M. Andrew Kachites, T. Sebastian, and M. Tom, "Text Classification from Labeled and Unlabeled Documents using EM," Mach. Learn., vol. 39, pp. 103-134, 2000.
[38] Ciao Lyao et Al., "Feature preparation in text categorization", Oracle Corporation
[39] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," Acm Computing Surveys, vol. 31, pp. 264-323, 1999.
[40] Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. (1992). Scatter/Gather: A Cluster-Based Approach to Browsing Large Document Collections. In Proceedings of the 15th Annual International ACM–SIGIR Conference on Research and Development in Information Retrieval. Copenhagen, ACM Press, New York: 318–329.
[41] Hearst, M. A., and Pedersen, J. O. (1996). Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In Proceedings of ACM SIGIR '96. Zurich, ACM Press, New York: 76–84.
[42] Tombros, A., Villa, R., and Rijsbergen, C. J. (2002). "The Effectiveness of Query-Specific Hierarchic Clustering in Information Retrieval." Information Processing & Management 38(4): 559–582.
[43] Liu, X., and Croft, W. B. (2003). "Statistical Language Modeling for Information Retrieval." Annual Review of Information Science and Technology 39.
[44] M. Spiliopouloux, F. Rinaldi, W. J. Blacky, and G. P. Zarriz, "Coupling Information Extraction and Data Mining for Ontology Learning in PARMENIDES," presented at RIAO2004, Vaucluse, France, 2004.
[45] Johan Bollen et Al. "Trend analysis of the digital library community", D-Lib magazine, January 2005, Volume 11, Number 1.
[46] Ronen Feldman et Al., "Mining biomedical literature using information extraction", Current Drug Discovery, Volume2, Issue 10, pages 19-23,October 2002.
[47] B. Settles (2005). ABNER: an open source tool for automatically tagging genes, proteins, and other entity names in text. Bioinformatics, 21(14):3191-3192.
[48] International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, BioNLP/NLPBA2004 SharedTask, http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm, online on April 2008
[49] BioCreAtIvE - Critical Assessment for Information Extraction in Biology challenge evaluation, http://biocreative.sourceforge.net/ online on April 2008
[50] Porter, 1980, An algorithm for suffix stripping, Program, Vol. 14, no. 3, pp 130-137
[51] C. D. Paice, "Another stemmer," SIGIR Forum, vol. 24, no. 3, pp. 56–61,1990
[52] Lovins,J.B. 1968: "Development of a stemming algorithm". Mechanical Translation and Computational Linguistics, 11, 22-31(1968).

ANEXOS

ANEXO A

Resultados da experiência SEQ_NO_COMP

25_TF_IDF

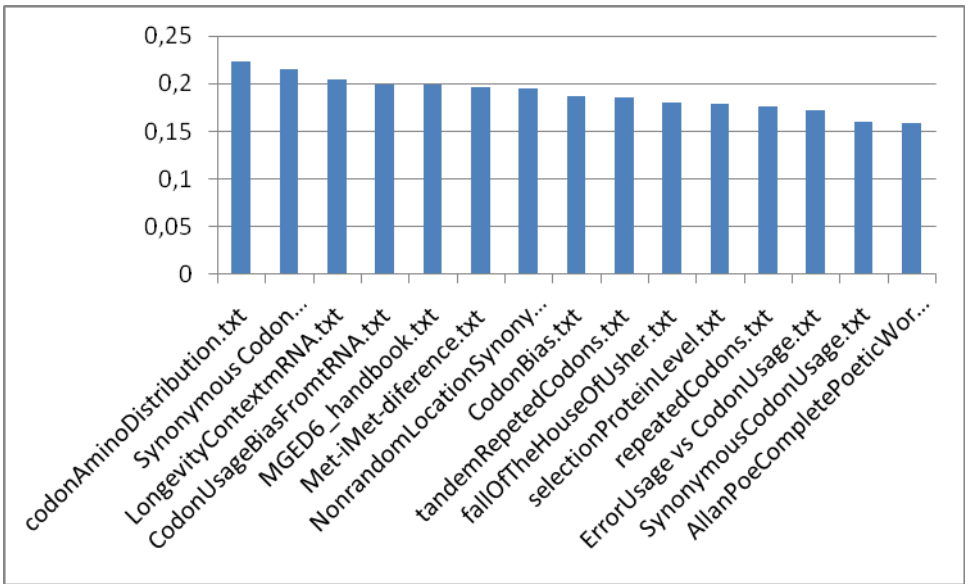

Figura 51 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso TF-IDF do corpus SEQ_ANALYSIS.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	12	2	0	1	15
Similaridade Média	0.191148	0.169522	0.0	0.198508	0.188755

Tabela 1 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/131=0.114
Recall	15/15=1
R-precision	12/15=0.8

Tabela 2 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

25_CTF_N

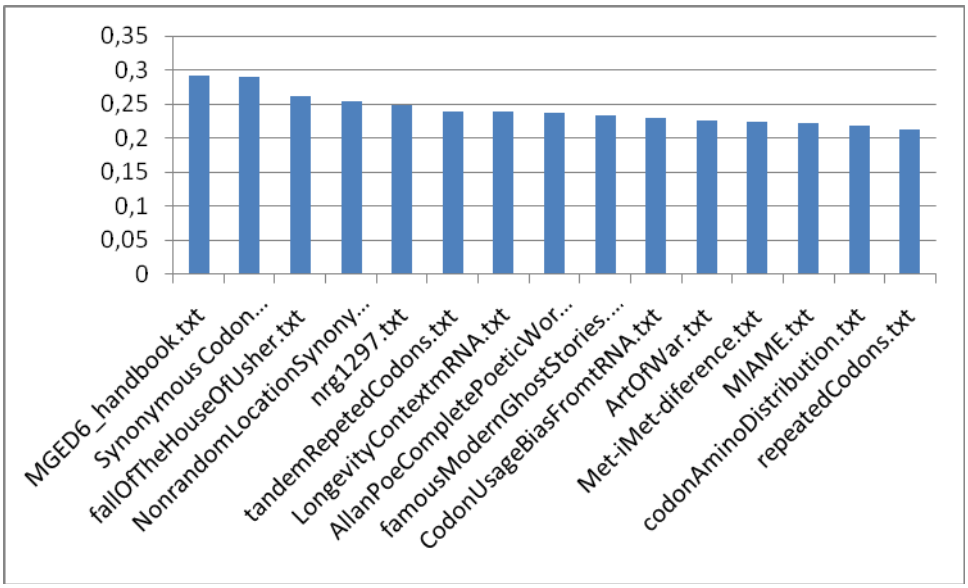

Figura 52 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso CTF_N do corpus SEQ_ANALYSIS.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	8	4	0	3	15
Similaridade Média	0.238511	0.239381	0.0	0.253505	0.241742

Tabela 3 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/131=0.114
Recall	15/15=1
R-precision	8/15=0.53(3)

Tabela 4 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

25_CTF

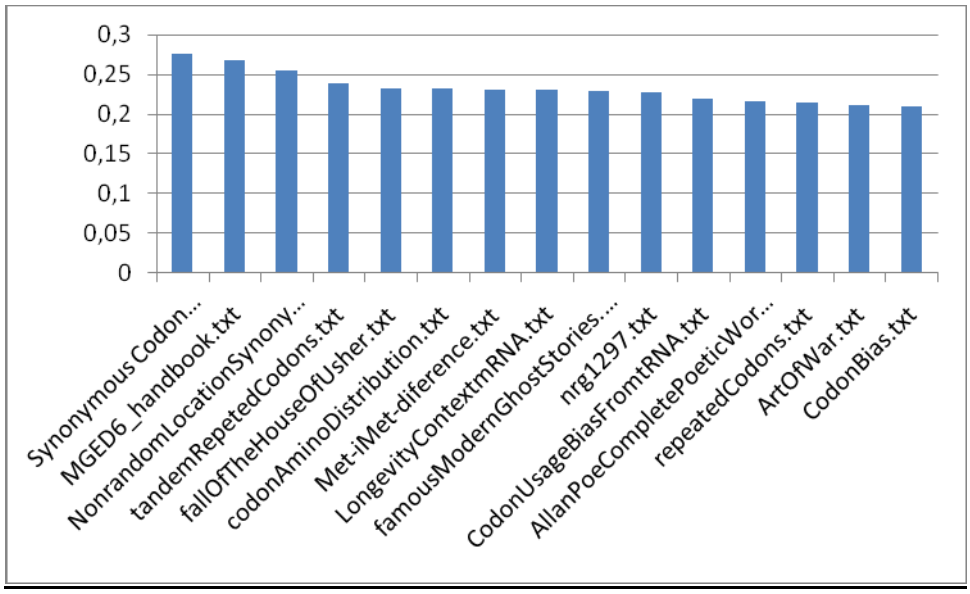

Figura 53 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso CTF do corpus SEQ_ANALYSIS.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	9	4	0	2	15
Similaridade Média	0.233097	0.222484	0.0	0.247167	0.232554

Tabela 7 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/131=0.114
Recall	15/15=1
R-precision	9/15=0.6

Tabela 8 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

100_TF_IDF

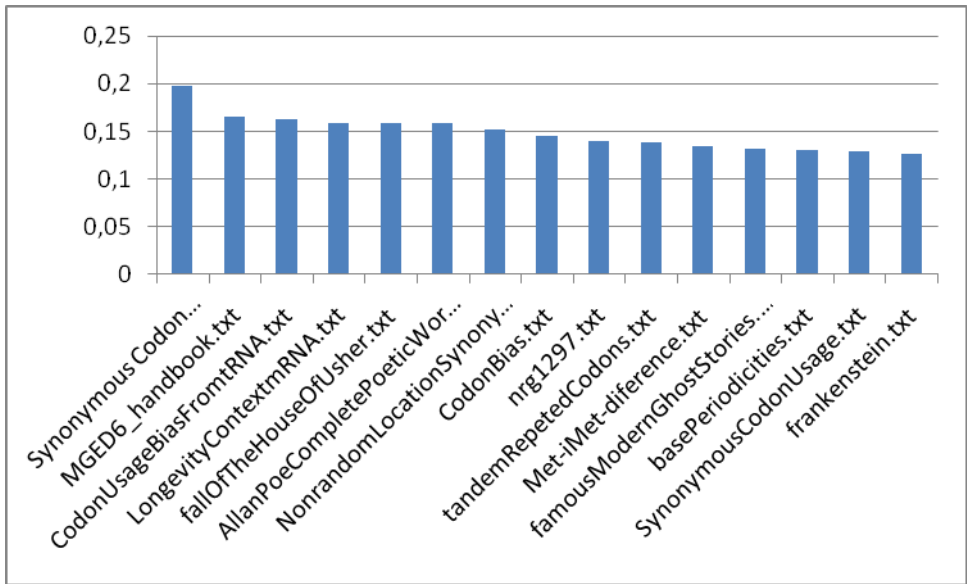

Figura 54 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta todos os termos com peso TF-IDF superior a zero do corpus SEQ_ANALYSIS.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	9	4	0	2	15
Similaridade Média	0.149588	0.143721	0.0	0.152282	0.148382

Tabela 9 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/131=0.114
Recall	15/15=1
R-precision	9/15=0.6

Tabela 10 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

100_CTF_N ou 100_CTF

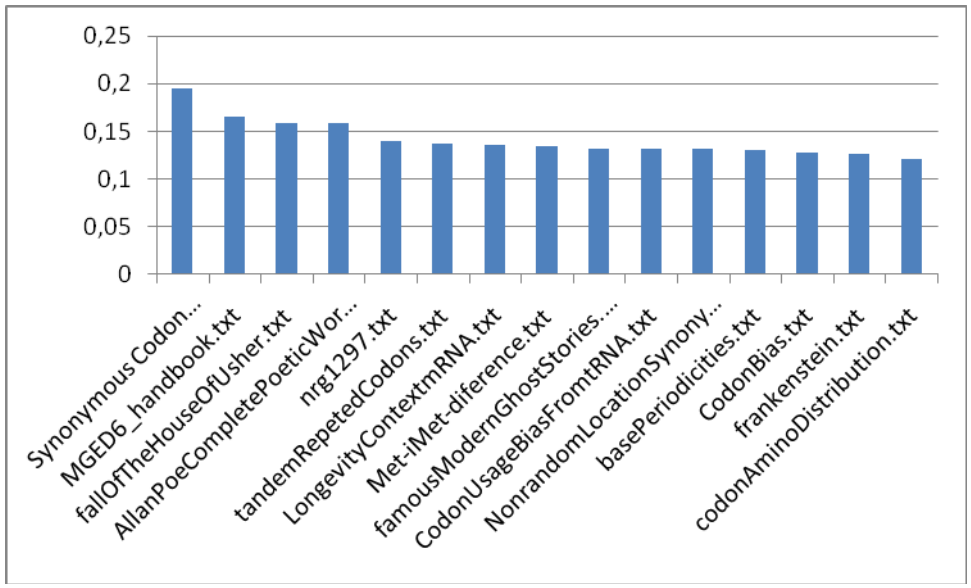

Figura 55 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta todos os termos com peso CTF ou CTF_N superior a zero do corpus SEQ_ANALYSIS.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	9	4	0	2	15
Similaridade Média	0.138239	0.143732	0.0	0.152276	0.141575

Tabela 11 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/131=0.114
Recall	15/15=1
R-precision	9/15=0.6

Tabela 12 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

ANEXO B

Resultados da experiência SEQ_COMP_GEN

25 TF IDF

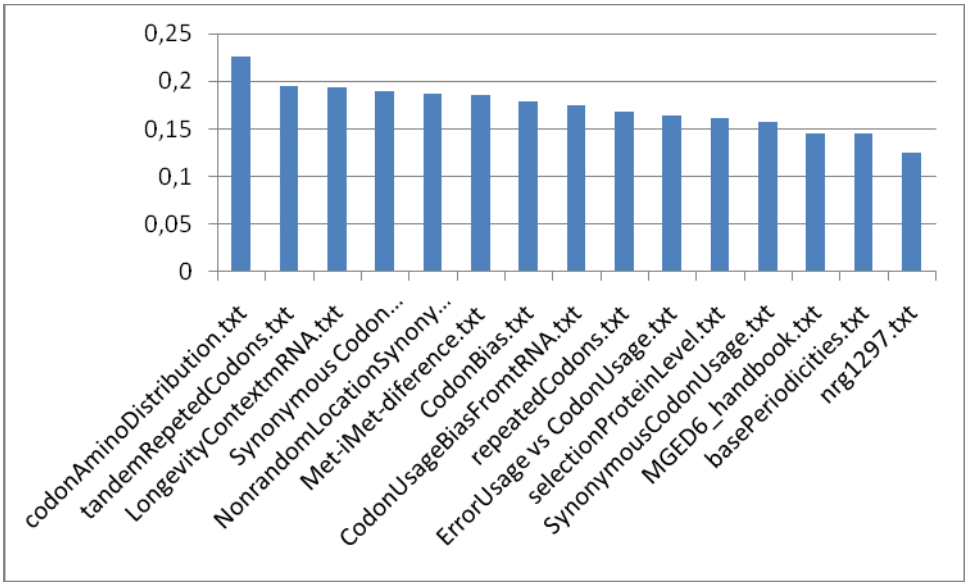

Figura 56 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso TF-IDF do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com o corpus GENERIC.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	13	0	0	2	15
Similaridade Média	0.178825	0.0	0.0	0.135369	0.173031

Tabela 14 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/131=0.114
Recall	15/15=1
R-precision	13/15=0.86(6)

Tabela 15 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

25_CTF_N

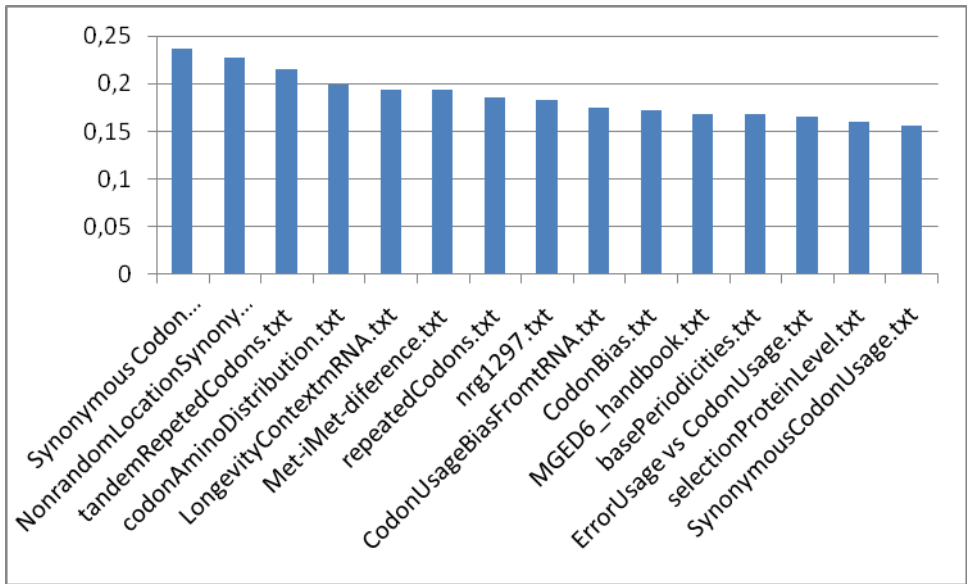

Figura 57 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso CTF_N do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com o corpus GENERIC.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	13	0	0	2	15
Similaridade Média	0.188128	0.0	0.0	0.175471	0.186441

Tabela 16 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/123=0.122
Recall	15/15=1
R-precision	13/15=0.86(6)

Tabela 17 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

25_CTF

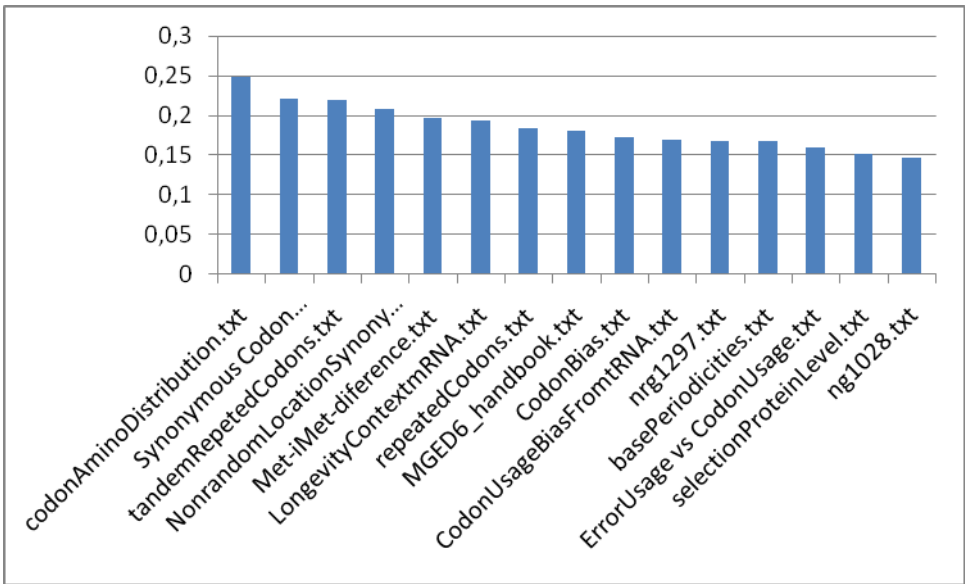

Figura 58 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso CTF do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com o corpus GENERIC.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	12	0	0	3	15
Similaridade Média	0.190999	0.0	0.0	0.164863	0.185772

Tabela 18 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/123=0.122
Recall	15/15=1
R-precision	12/15=0.8

Tabela 19 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

100 TF_IDF

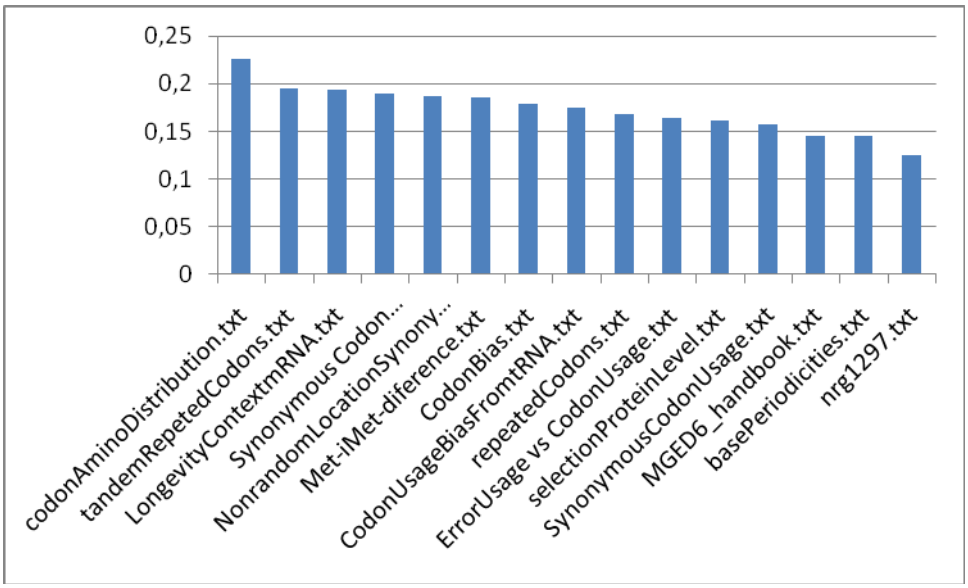

Figura 59 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os termos com peso TF-IDF maior que zero, do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com o corpus GENERIC.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	13	0	0	2	15
Similaridade Média	0.178825	0.0	0.0	0.135369	0.173031

Tabela 20 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/131=0.114
Recall	15/15=1
R-precision	13/15=0.86(6)

Tabela 21 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

100_CTF_N ou 100_CTF

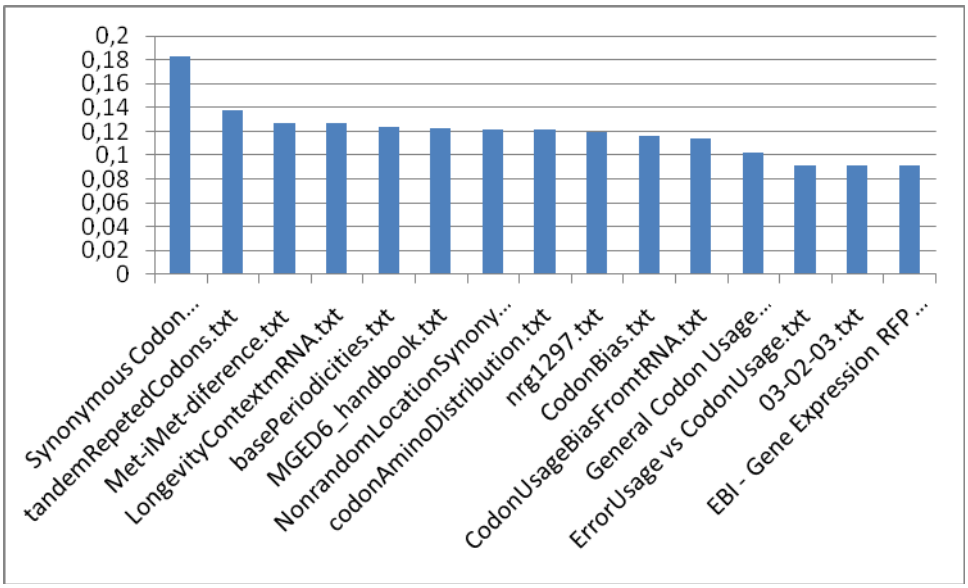

Figura 60 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta todos os termos com peso CTF ou CTF_N superior a zero do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com o corpus GENERIC.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	11	0	0	4	15
Similaridade Média	0.124120	0.0	0.0	0.106228	0.119349

Tabela 22 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/123=0.122
Recall	15/15=1
R-precision	11/15=0.73(3)

Tabela 23 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

ANEXO C

Resultados da experiência SEQ_COMP_MA_NCB

25 TF_IDF

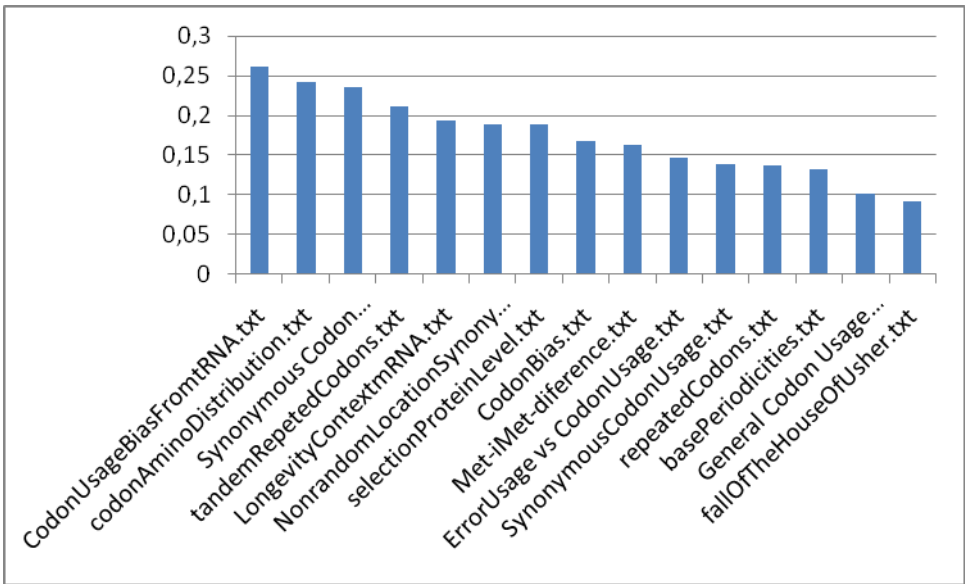

Figura 61 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso TF-IDF do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA e NCB.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	14	1	0	0	15
Similaridade Média	0.179247	0.09148	0.0	0.0	0.173396

Tabela 25 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/22=0.636
Recall	14/15=0.93(3)
R-precision	14/15=0.93(3)

Tabela 26 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

25_CTF_N

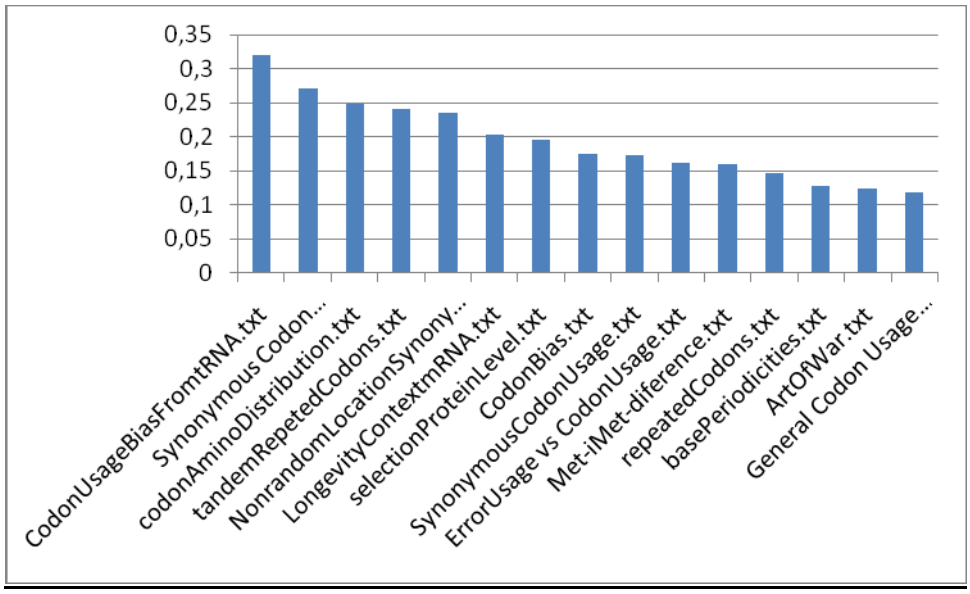

Figura 62 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso CTF_N do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA e NCB.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	14	1	0	0	15
Similaridade Média	0.198629	0.123459	0.0	0.0	0.193617

Tabela 27 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/22=0.636
Recall	14/15=0.93(3)
R-precision	14/15=0.93(3)

Tabela 28 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

25_CTF

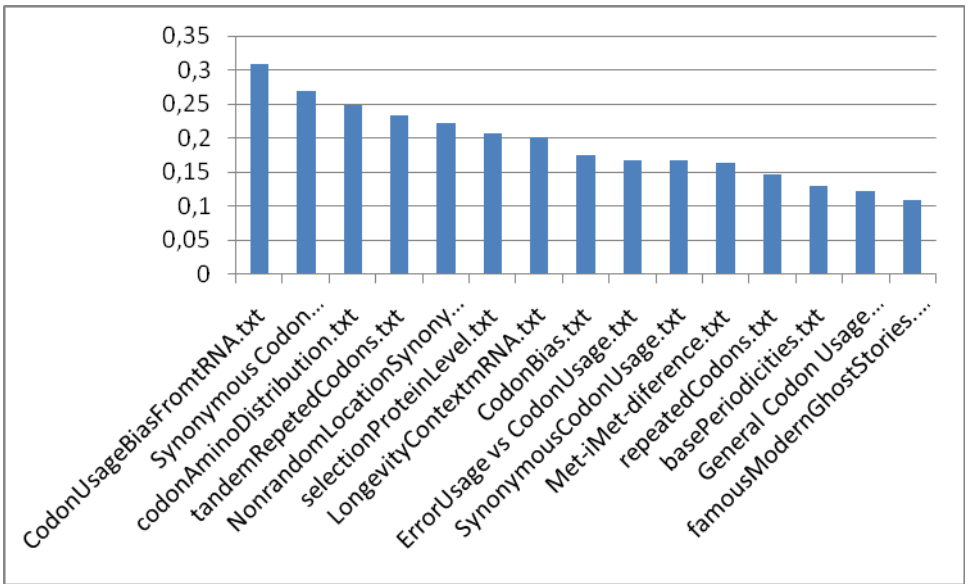

Figura 63 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso CTF do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA e NCB.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	14	1	0	0	15
Similaridade Média	0.197207	0.109390	0.0	0.0	0.191352

Tabela 29 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/22=0.636
Recall	14/15=0.93(3)
R-precision	14/15=0.93(3)

Tabela 30 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

100 TF_IDF

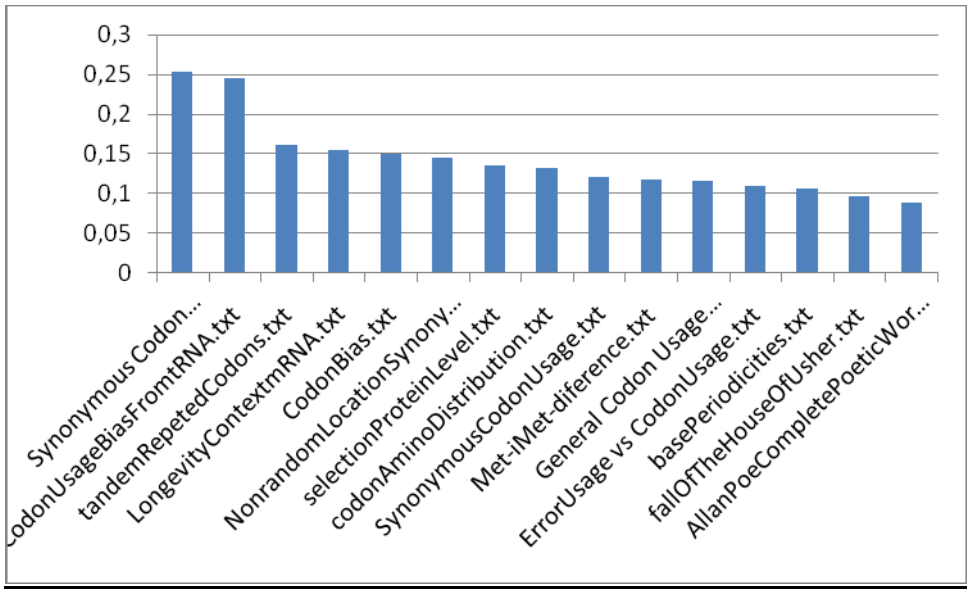

Figura 64 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os termos com peso TF-IDF maior que zero, do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA e NCB.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	13	2	0	0	15
Similaridade Média	0.149856	0.092335	0.0	0.0	0.142187

Tabela 31 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/22=0.636
Recall	14/15=0.93(3)
R-precision	13/15=0.86(6)

Tabela 32 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

100_CTF_N ou 100_CTF

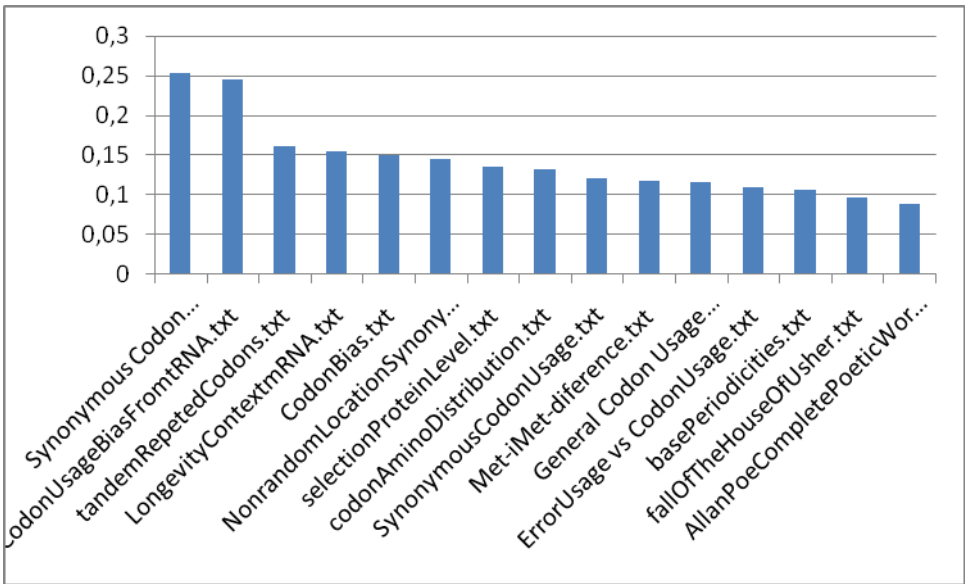

Figura 65 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta todos os termos com peso CTF ou CTF_N superior a zero do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA e NCB.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	13	2	0	0	15
Similaridade Média	0.149856	0.092335	0.0	0.0	0.142187

Tabela 33 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/22=0.636
Recall	14/15=0.93(3)
R-precision	13/15=0.86(6)

Tabela 34 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

ANEXO D

Resultados da experiência SEQ_COMP_ALL

25 TF_IDF

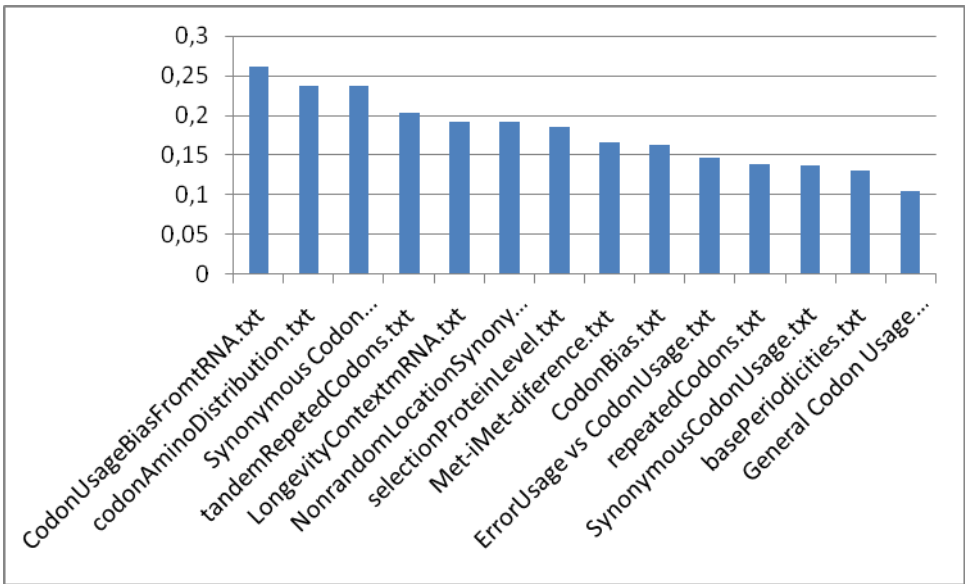

Figura 66 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso TF-IDF do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA, NCB e GENERIC.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	14	0	0	0	14
Similaridade Média	0.178331	0.0	0.0	0.0	0.178331

Tabela 36 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/14=1
Recall	14/15=0.93(3)
R-precision	14/15=0.93(3)

Tabela 37 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

25_CTF_N

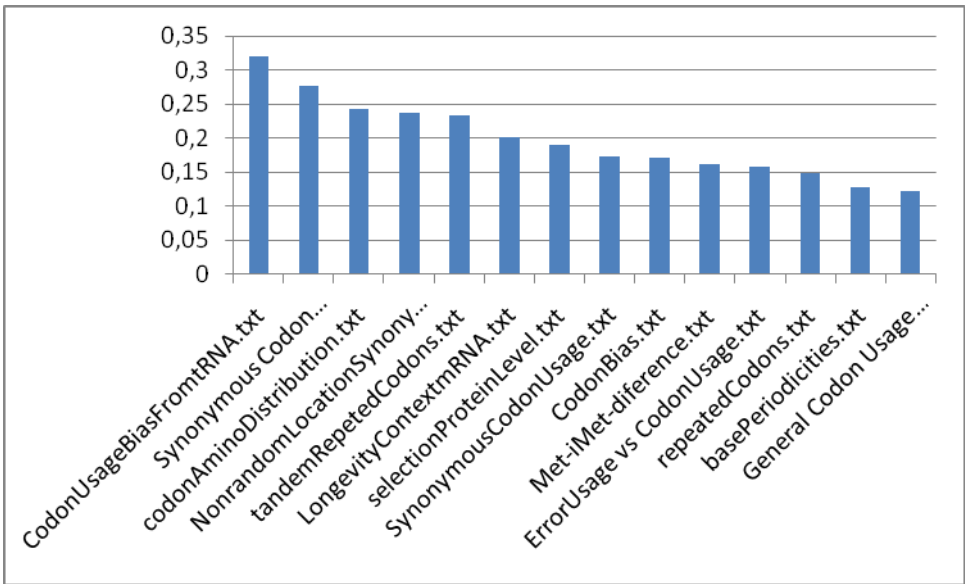

Figura 67 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso CTF_N do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA, NCB e GENERIC.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	14	0	0	0	14
Similaridade Média	0.197267	0.0	0.0	0.0	0.197267

Tabela 38 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/14=1
Recall	14/15=0.93(3)
R-precision	14/15=0.93(3)

Tabela 39 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

25_CTF

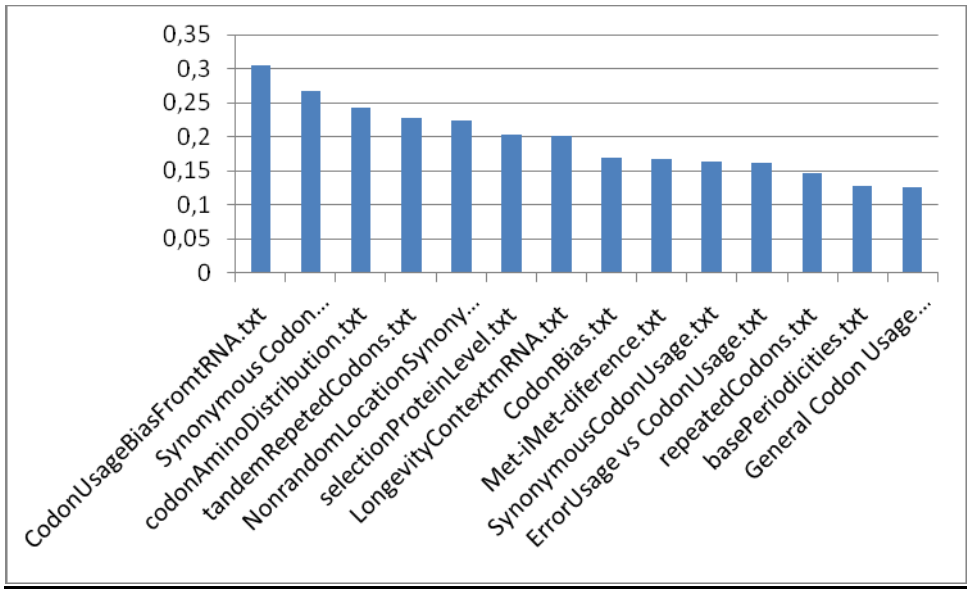

Figura 68 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os 25 por cento dos termos com maior peso CTF do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA, NCB e GENERIC.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	14	0	0	0	14
Similaridade Média	0.195322	0.0	0.0	0.0	0.195322

Tabela 40 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/14=1
Recall	14/15=0.93(3)
R-precision	14/15=0.93(3)

Tabela 41 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

100 TF_IDF ou 100_CTF_N ou 100_CTF

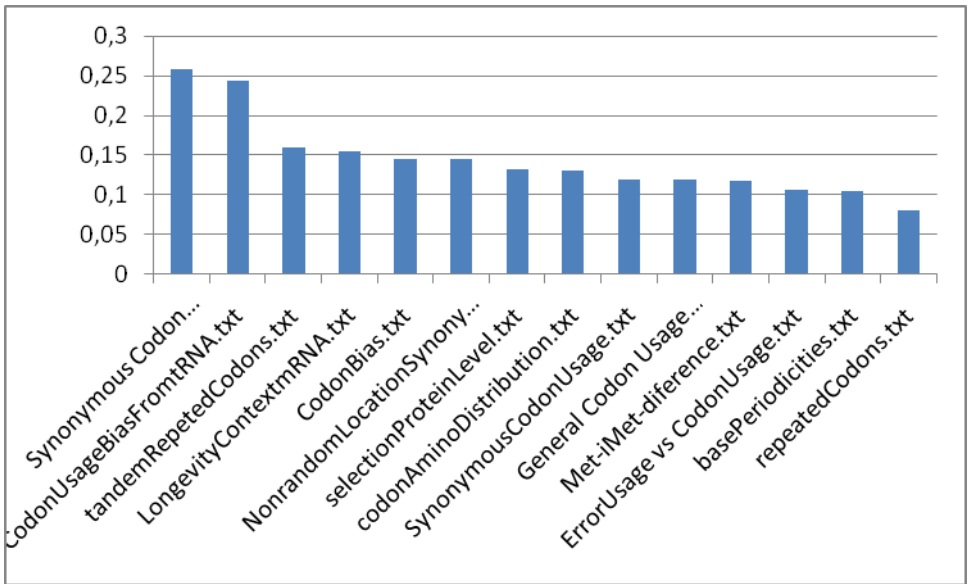

Figura 69 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os termos com pesos TF-IDF, CTF ou CTF_N maiores que zero, do corpus SEQ_ANALYSIS, após remoção no sistema Corparator dos termos em comum com os corpus MADATA e NCB.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	14	0	0	0	14
Similaridade Média	0.144070	0.0	0.0	0.0	0.144070

Tabela 42 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	14/14=1
Recall	14/15=0.93(3)
R-precision	14/15=0.93(3)

Tabela 43 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.

ANEXO E

Resultados da experiência SEQ_ANNO_FIL

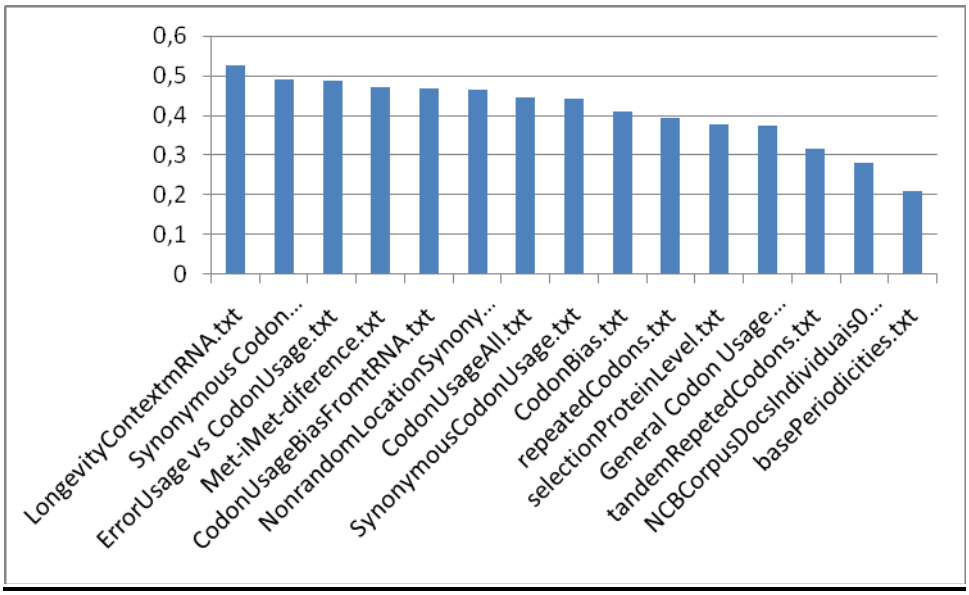

Figura 70 – Gráfico dos primeiros 15 resultados de similaridade no sistema BDClassifier, utilizando como consulta os termos do corpus SEQ_ANALYSIS resultantes da aplicação de anotação e filtragem utilizando as capacidades de reconhecimento de entidades da ferramenta ABNER.

	SEQ_ANALYSIS	GENERIC	NCB	MADATA	TOTAL
Nº Documentos	14	0	1	0	14
Similaridade Média	0.419250	0.0	0.279465	0.0	0.409931

Tabela 45 – Presença de documentos dos diversos corpus de teste nos primeiros 15 resultados e similaridades médias totais respectivas.

Medida	Resultado
Precision	15/131=0.114
Recall	15/15=1
R-precision	14/15=0.93(3)

Tabela 46 – Resultados obtidos de Precision e Recall e fracção de documentos SEQ_ANALYSIS presentes nos primeiros 15 de maior similaridade.